The Dynamic Modelling and Control System of a Tethered Aerostat for Remote Sensing Applications

by

Daniël Andries Fourie

Thesis presented in partial fulfilment of the requirements for the degree Master of Science in Engineering at the Department of Electrical and Electronic Engineering at the University of Stellenbosch

Department of Electrical Engineering,
University of Stellenbosch,
Private Bag X1, 7602 Matieland, South Africa.

Supervisor: Prof W.H. Steyn

December 2009

Declaration

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the owner of the copyright thereof (unless to the extent explicitly otherwise stated) and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Date: December 2009

Abstract

Aerostats and Stratolites could play a major role in expanding current satellite and other technologies in the near future. A study was made on the development of aerostat platforms and the current state of Stratolite development.

The aim was to develop an airship system that is capable of maintaining a specific position regardless of the presence of wind. The various applications of such a geostationary platform are discussed.

A dynamic model of an airship was developed and a simulation was implemented in software. This was done to study the possibility of developing aerostats like these.

A tethered airship system was developed and built to demonstrate that it is possible to control the position of an airship. The airship system uses current technology in an unique combination to fulfil the requirement of remaining stationary despite the influence of wind.

Various control system design techniques were used to implement the controllers. Linear models of the airship system were identified practically and used to design the controllers.

The controllers were tested in simulation as well as practically and the results of these tests are given. It was concluded that there exists potential for the development of Stratolite systems, although there exists a fair amount of challenges and obstacles that would need to be overcome before this technology could be implemented.

Uittreksel

Aerostats en Stratolites kan 'n besondere rol speel in die uitbreiding van huidige sateliet- en ander aardwaarnemingstoepassings. 'n Studie is gemaak oor die ontwikkeling van Aerostat platforms en die huidige stand van Stratolite ontwikkeling.

Die mikpunt was om 'n lugskipstelsel te ontwikkel wat in staat is om 'n spesifieke posisie te handhaaf ten spyte van die invloed van wind. Die verskeidenheid van toepassings, waarvoor so 'n geostasionêre platform gebruik kan word, word genoem.

'n Dinamiese model van 'n lugskip is ontwikkel en die stelsel is in sagteware gesimuleer. Dit is gedoen om die moontlikheid te ondersoek om sulke Aerostats in die toekoms te ontwikkel.

'n Lugskipstelsel, wat aan die grond geanker word met 'n kabel, is ontwerp en gebou. Die stelsel is gedemonstreer en daar is bewys dat dit moontlik is om die posisie van die lugskip te beheer. Die lugskip gebruik huidige tegnologie wat in 'n unieke kombinasie saamgevoeg is om te illustreer dat dit moontlik is vir die lugskip om stasionêr te bly ten spyte van wind.

Verskeie beheerstelsels ontwerptegnieke is gebruik om die beheerders mee te implementeer. Lineêre modelle van die lugskip is prakties geïdentifiseer en is gebruik om die beheerders te ontwerp.

Die lugskip se beheerders is in simulasie sowel as prakties getoets en die resultate van hierdie toetse word gegee. Die projek bevestig dat daar 'n potensiaal bestaan vir die praktiese ontwikkeling van Stratolite stelsels. Daar is egter 'n hele paar uitdagings en probleme wat eers uit die weg geruim sal moet word, voordat hierdie tegnologie 'n alledaagse werklikheid sal word.

Acknowledgements

I would like to sincerely thank the following persons, without whom this project would not be completed:

- Prof. W.H. Steyn, for his leadership and guidance for the duration of the project. Thank you for all the advice and allowing me to learn more about engineering.
- Mr. G. Avenant, who worked on the sister-project of this project, [1]. Thank you for all your assistance before, during and after flight tests.
- Mr. J. Arendse and Mr. Q. Brandt, for all your assistance in the laboratory, especially with soldering circuit boards and ordering bottles of hydrogen and helium.
- Mr. W. van Rooyen, for your help in designing and building the gondola.
- Mr. P. Petzer, for letting me store the airship in the workshop and helping with the setup procedures prior to test flights.
- Mr. A.M. de Jager, Mr. C. Jaquet and all the other ESL students who gave advice and support during the last two years.
- I would like to thank everybody in the ESL for making the last two years a memorable experience. Good luck with your careers. I hope you will always be passionate about the work you are doing.
- Ms. Micaela Sawyer for proof reading this thesis.
- Finally, I would like to thank my parents for giving me the opportunity to study. Thank you for believing in me and praying for me.

Contents

List of Figures

List of Tables

Nomenclature

Acronyms

ADC	Analog to Digital Converter
ATG	Advanced Technology Group
CFD	Computational Fluid Dynamics
COM	Centre of mass
COV	Centre of volume
CSV	Comma Separated Values
DC	Direct Current
DOF	Degrees of Freedom
ESA	European Space Agency
ESC	Electronic Speed Controller
GHz	Gigahertz
GPS	Global Positioning System
GUI	Graphical User Interface
HAP	High Altitude Platform
HTA	Heavier than Air
Hz	Hertz
IMU	Inertial Measuring Unit
kg	Kilogramme
kbps	Kilobits per second
kHz	Kilohertz
LED	Light Emitting Diode
LTA	Lighter than Air
m^3	Cubic meters
ms	Milliseconds
mV	Millivolt
m/s	Meters per second
MHz	Megahertz
N	Newton
Nm	Newton meter
PL	Pseudolite
PVC	Polyvinyl chloride
PWM	Pulse Width Modulator
SPF	Stratospheric Platform
UART	Universal Asynchronous Receiver/Transmitter
UAS	Unmanned Aircraft Systems
V	Volt

Symbols

b	Fin semi span
B	Buoyancy force
C_D,C_L	Drag and Lift coefficients
C_{DC}	Cross-flow drag coefficient on the hull
d	Distance between brushless DC motors
$D(s)$	Transfer function of continuous compensator
$D(z)$	Discrete transfer function of digital compensator
D	Maximum cross-sectional diameter of the airship
g	Acceleration of gravity
$G(s)$	Transfer function of linearized airship model
$G(z)$	Discrete transfer function of linearized airship model
K	Open loop gain of airship model
k_1,k_2	Added-mass factors
k',k_{44}	Added mass factors
L	Length of the airship
L_{tether}	Length of the tether
L_f	Net upward lift force
L_d	Excessive lift force
L_{He_2}	Excessive lift force when using helium
L_{H_2}	Excessive lift force when using hydrogen
m	Total mass of the airship
m_{hull}	Mass of the hull
m_{fins}	Mass of the fins
m_{H_2}	Mass of hydrogen
$m_{gondola}$	Mass of gondola
m_{tether}	Mass of tether
p	Roll rate
q	Pitch rate
q_0	Dynamic pressure
r	Yaw rate
R	Body cross-sectional radius
S	Body cross-sectional area
S_F	Area of one fin
S_H	Surface area of the airship
T_{motor1}	Thrust force of motor 1
T_{motor2}	Thrust force of motor 2
T_s	Sampling time
t_s	Settling time
u	Translational velocity in direction of x-axis
$u_1(k)$	Control signal output of heading controller
$u_2(k)$	Control signal output of position controller
$u_{motor_1}(k)$	Control signal output that is commanded to motor 1
$u_{motor_2}(k)$	Control signal output that is commanded to motor 2
v	Translational velocity in direction of y-axis
$\dot{\mathbf{v}}$	First derivative of velocity

V_H	Volume of the hull
w	Translational velocity in direction of z-axis
W	Weight of airship including gas
W_0	Weight of the hull and fins only
x_{fs},x_{fe}	X-coordinates of the start and end positions of the fins
X,Y,Z	Coordinates of body frame origin in the inertial frame
x,y,z	Coordinates of a point in the body frame
x_I,y_I,z_I	Coordinates of a point in the inertial frame
z_{CL}	Closed loop pole positions
α	Angle of attack
β	Sideslip angle
$\Delta C_{p\alpha}$	Pressure coefficient of the fins
ϵ	Longitudinal distance from the nose
ϵ_0	Longitudinal position at which the flow ceases to be potential
ϵ_1	Longitudinal position at which the area of the hull decrease most rapidly
ϵ_m	Distance from the centre of volume to the nose of the airship
η	Efficiency factor
γ	Angle between centerline and velocity vector
ω_n	Natural frequency
ϕ,θ,ψ	Euler angles for roll, pitch and yaw
ρ_{air}	Air density
ρ_{gas}	Density of gas
ρ_{H_2}	Density of hydrogen
ρ_{He_2}	Density of helium
Θ	Angle between inertial x-axis and body x-axis
ζ	Damping coefficient
z_{CL}	Position of closed loop poles

Matrices

$\mathbf{A}^T$	Transformation matrix to transform from body coordinates to inertial coordinates
$\mathbf{F}$	Force vector
$\boldsymbol{F_G}$	Gravitational force
$\boldsymbol{F_{NV}}$	Force normal to the centerline due to viscous effects
$\boldsymbol{F_v}$	Force due to viscous effects in vector form
$\boldsymbol{F_{thrust}}$	Force generated by actuator
$\boldsymbol{I}_{3\times3}$	3×3 Identity matrix
$\mathbf{J}$	Second moment of inertia or inertia tensor
$\mathbf{M}$	Moment vector
$\boldsymbol{M_A}$	Added-mass matrix
$\boldsymbol{M_{AH}}$,$\boldsymbol{M_{AF}}$	6×6 added-mass matrix of the hull and fins respectively
$\boldsymbol{M_{rigid}}$	6×6 added-mass matrix of the airship
$\boldsymbol{M_{yaw}}$	Yaw-moment
$\boldsymbol{M_v}$	Moment due to viscous effects in vector form
$\boldsymbol{r_g}$	vector from centre of volume to centre of gravity
$\boldsymbol{r_g^\times}$	skew symmetric matrix of $\boldsymbol{r_g}$
$\boldsymbol{v}^T$	Translational velocity vector
$\boldsymbol{v_0}$	Velocity vector
$\dot{\mathbf{v}}$	First derivative of velocity
$\boldsymbol{\omega^T}$	Angular velocity vector
$\boldsymbol{\tau}$	Forces and moments vector

Chapter 1

Introduction

1.1 Background

There is a growing interest towards using unmanned, autonomous flight vehicles as high altitude platforms for military or commercial applications. These applications include Remote Sensing, Telecommunication, Surveillance and Enhanced Navigation to name just a few.

A high altitude platform (HAP) is sometimes referred to as a near-space platform, as it operates in the stratosphere, between $10km$ and $50km$ above sea level. These high altitude platforms could potentially be less expensive than current space-based solutions. Near-space platforms have another advantage above space-based systems in that it is constantly above a certain location, unlike non-geostationary satellites which only pass over that location occasionally. It is however difficult to design a near-space vehicle capable of remaining aloft for extended periods of time, but this characteristic actually allows near-space vehicles with another advantage: The capability to descend back to the ground for maintenance purposes, something that is not possible with satellites.

Two types of vehicles can be considered for the purpose of a HAP, namely lighter-than-air (LTA) vehicles or heavier-than-air (HTA) vehicles. An airship is a LTA aircraft with propulsion and steering systems. Unlike conventional HTA vehicles such as aeroplanes and helicopters whose lift is aerodynamically generated by moving an aerofoil through the air, airships stay aloft using a light lifting gas. This distinguishing feature provides LTA vehicles with long endurance, a high payload-to-weight ratio and low fuel consumption. [7]

This thesis looks at the possibility of using an airship as a HAP. A HAP that uses an airship as vehicle is called a Stratolite.

This thesis explains the procedure of developing a control system that controls the position of an airship and shows that it is possible to use an airship as a stationary platform, that is capable of holding its position despite the presence of wind. The airship's ability to remain stationary over time is tested at low altitudes of up to $20m$.

In this thesis a tether is fitted to the airship to keep the altitude fixed. When implementing a HAP practically, no tethers will be used. In such a case the altitude of the airship varies according to the change in temperature and air density. Special design techniques must be used in order for an airship to be capable of reaching a high altitude in the stratosphere.

When an airship is filled with a LTA gas, such as Helium, and the airship is released into the atmosphere, the airship will start ascending. The density of the air surrounding the airship will

decrease with an increase of altitude. The density of the gas inside the airship's hull also decreases. This entails that the gas expands inside the hull. In the case of a weather balloon, the balloon will expand together with the gas until the balloon eventually bursts. If the airship's hull is rigid and can't expand beyond a certain point, the airship will in effect become heavier than the surrounding air and the airship's altitude will stop increasing. This is called the 'pressure height' or 'ceiling altitude'. This ceiling altitude would alter with changes in temperature, for example between day and night. The effect of temperature changes could be described as follows: When the temperature increases, the density of the gas inside the hull decreases and the volume of the gas in the hull expands. When the temperature decreases, the density of the gas inside the hull increases and the volume of the gas in the hull contracts. The change in density with altitude has a more pronounced effect than the change of density due to temperature. [8]

The airship is required to reach very high altitudes. Conventional airships are only capable of reaching altitudes of $2km$, according to [9], which is much lower than the required $20km$ for stratospheric use. Special design techniques need to be made to ensure that the airship is able to operate at the required altitude. In general there exist two types of airship designs. These are rigid airships and pressurised airships. The differences of these airship designs are described in [10]. Computational fluid dynamics (CFD) tools have recently been developed and used to obtain optimal geometries for airships to minimise drag and turbulence effects. According to [10] the first step of designing an airship is to obtain the correct size of the hull. The volume of the hull defines the buoyant lift capability of the airship, and determines the maximum attainable altitude. The materials that are used for the hull must be chosen wisely because of the influence that the mass, volume, and gas densities have on the aerostatic principles. Next, with the size and shape of the airship defined, we may compute the drag force experienced at a particular airspeed. This in turn drives the thrust and power requirements.

The airship should be able to stay in the stratosphere for extended periods of time. The effectiveness of a HAP is directly related to its endurance and therefore it is crucial to design for the best possible endurance. There are multiple design parameters that will influence the endurance of an airship. These parameters include things such as the power available on the airship, the power needed to remain stationary and the airship's ability to contain the LTA gas.

The next section contains valuable information on previous research which has been done on using airships as suitable vehicles for HAPs.

1.2 Literature Review

1.2.1 Airship Technology

Airships have been around for quite some time, being used for anything from bombers in the First World War and intercontinental passenger carriers, to merely floating above a building while displaying the name of a company for advertising purposes. After the Hindenburg disaster in 1936, the airship industry lost some credibility. The development of conventional aircraft like aeroplanes and helicopters soon diminished the need for airships, until recently:

The development of modern techniques, such as composite materials, optimal design, computational fluid dynamics (CFD), thermal modeling and automatic control, brought a resurgence of these

aircraft. [3]. Since the resurgence a wide range of applications have been proposed for airships, such as surveillance, advertising, aerial photography, monitoring and research as described by Hima [11]. With the growing interest regarding unmanned aerial vehicles, a demand for controllable unmanned airships has surfaced. In [12] and [8] the benefits of using airships are described:

Compared to other aerial vehicles, airships have very long endurance. Airships do not need a lot of energy to stay afloat, only to manoeuvre. The relatively small engines used on airships don't produce much noise and turbulence, resulting in minimal environmental disturbance. Sensor noise is reduced due to low vibration. Airships have low radar and infrared signatures, which is ideal for military applications. [3]

Because of the growing interest concerning airships, a fascinating amount of development and research on airship relative fields have emerged. Companies, like 21st Century Airships Inc. [9], specialises in designing airships that are capable of reaching very high altitudes, while the company TCOM develops advanced tethered aerostat systems for surveillance and communication purposes, as described in [13]. Their airship systems operate at altitudes of up to $5km$ and are also connected to the ground with a tether. Figure 1.1 shows an example of one of TCOM's aerostats, while Figure 1.2 shows a graphical illustration of how the tethered aerostat is used to monitor the activity of drill ships, and allows communication between different drill ships, from the shore.

Figure 1.1: TCOM Aerostat

The University of California has done some research on using airships for exploration purposes on other planets in our solar system, as described in [14].

There are currently two airship related projects in progress at the University of Stellenbosch: The project concerning this thesis is one of them, while the other project involves an automated flight control system for an airship, as described in [1].

These are not the first airship related projects that are attempted at the University of Stellenbosch: Bijker [15] finished a project on the development of an attitude heading reference system for an airship in 2006. The current projects are based on his findings.

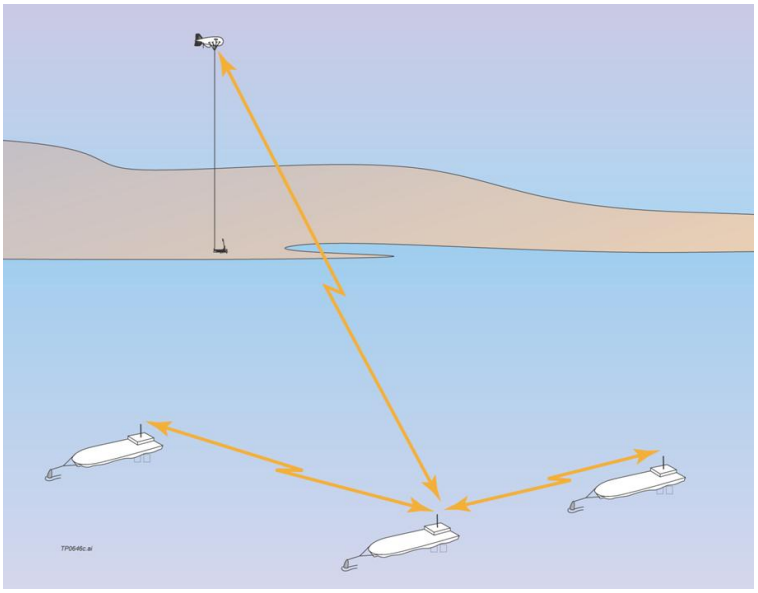


Figure 1.2: Application of TCOM Aerostat

1.2.2 High Altitude Platforms and Aerostats

According to [7], there has been a growing interest during the last two decades for developing autonomous atmospheric flight vehicles as platforms operating for extended periods of time at very high altitudes (between $20km$ and $50km$). These high altitude platforms (HAPs) could accomplish military and commercial missions previously accomplished using spacecraft. One of the primary advantages of a HAP above any satellite is that it is recoverable. This allows HAPs to be maintained frequently and even upgraded between missions.

Figure 1.3 shows an example of a heavier-than-air HAP that was developed by AeroVironment, Inc.

Figure 1.3: Global Observer vehicle from AeroVironment

AeroVironment, Inc. is a Californian based company that develops unmanned aircraft systems

(UAS), among other things. These include the Raven, Wasp, Puma AE and the Dragon Eye as seen on AeroVironment's website [16]. They are one of the leading developers of HAP technologies. The vehicle in Figure 1.3 is called the Global Observer. It is capable of reaching altitudes of $20km$, and has a superb station keeping ability, even in strong winds. It can stay aloft for up to one week, which is a long time for a HTA vehicle. The Global Observer is powered with liquid hydrogen and emits no carbon. [17]

Airships, in contrast with the Global Observer, are LTA vehicles which are capable of much longer flight missions than just one week. This is what makes airships such valuable vehicles to use as HAPs. Figure 1.4 shows graphically how airships in the stratosphere could be used in combination with satellites. This figure illustrates how airships can be used as Stratospheric Platforms (SPFs) in combination with GPS satellites. The airships are fitted with GPS-like transmitters called pseudolites (PL) to enhance the performance of GPS. [18]

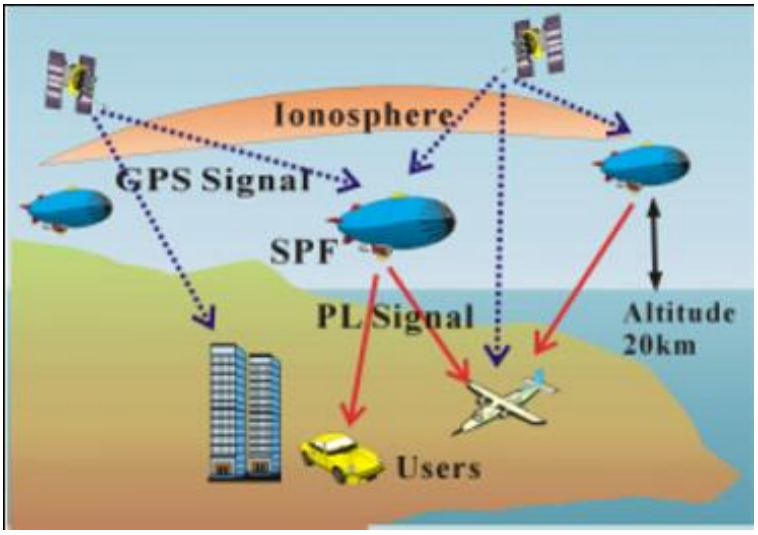


Figure 1.4: Airships in the stratosphere

One of the major challenges of HAPS is to ensure that the HAP has an effective station keeping ability. Winds in the stratosphere are relatively modest, as the stratosphere is high above the jet streams. This is shown in Figure 1.5. This thesis specifically aims to design a control system that flies the airship directly into the wind and remain stationary by doing so.

Lindstrand Balloons is a world renowned manufacturer of LTA vehicles in the UK. In December 1998 the European Space Agency (ESA) awarded Lindstrand Balloons a design study contract for a geostationary stratospheric unmanned airship. In an effort to investigate the propagation characteristics at an altitude of 25 kilometres, Lindstrand Balloons built a $14,000m^3$ super-pressure airship that carried a $47GHz$ test transmitter. This airship is unmanned and can remain geostationary in the stratosphere with a $600kg$ payload. According to Lindstrand Balloons, the goal is to achieve mission times of two to five years. [2]

According to [19], Japan has also been developing a similar airship system for more than a decade.

StratSat is a project by the Advanced Technology Group (ATG). They want to use airships as Stratolites to replace the terrestrial towers that are put up by mobile phone companies. StratSat

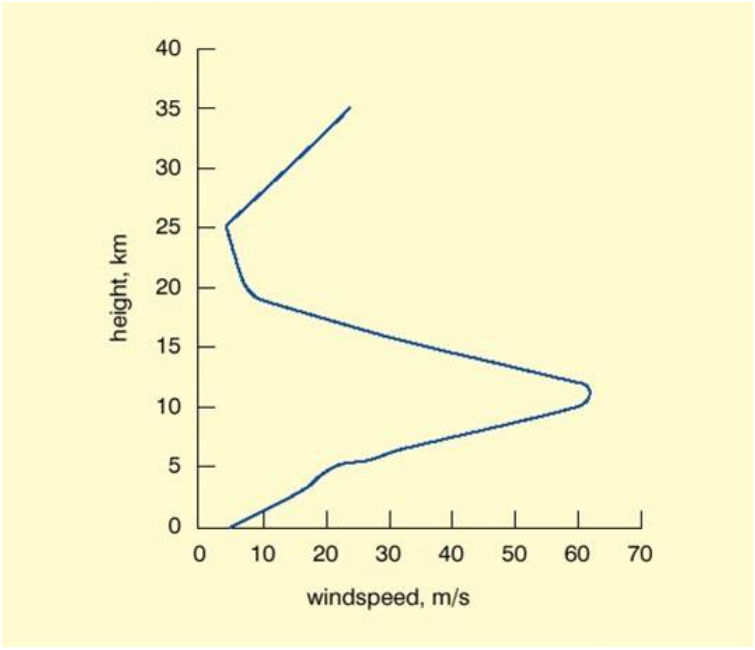


Figure 1.5: Average wind velocity at different altitudes (Source: [2])

could offer a highly cost effective cellular service, broadband data communications, local area television and very high speed Internet access without the need for extensive and costly terrestrial towers. ATG believes it could cover an area the size of England with only 19 StratSats, eliminating the need to build and maintain 10,000 terrestrial towers. [2]

A project at the University of Pennsylvania envisions an autonomous airship that can carry environmental sensors safely and accurately through a specific airspace. Using solar power as an energy source, an airship will be able to operate over an extended period of time, allowing it to collect data over large geographical areas or large volumes of airspace. A sensing platform of this kind would be an asset to environmental scientists who seek to understand and manage the environment. [20]

All these projects serve as proof that there exist a large potential for modern airship technologies. This thesis aims to exploit this potential, and get a little bit closer to developing a sustainable geostationary Stratolite platform in the form of an airship.

1.2.3 Airship Dynamics

The resurgence of airships has created a need for dynamic models and simulation capabilities for airships to be developed. In most dynamic models of aircraft, the vehicles are modelled as a rigid body with three translational and three rotational degrees of freedom (DOF). These dynamic models can be represented by six differential equations, which have been derived in several textbooks for conventional aircraft, such as the one by Etkin [21]. However, the differences between HTA and LTA aircraft, particularly with regards to the buoyancy of LTA aircraft and those related to the inertia of the surrounding air, imply that models specific to airships must be developed.

In Yuwen Li's thesis [3], a dynamics model of a rigid-body airship is presented. Li elaborates on the structural dynamics, aerostatics, aerodynamics, and flight dynamics of flexible airships as well. A comprehensive aerodynamic computational approach is applied, where the aerodynamic

forces and moments are categorised into various terms based on their physical effects on the airship. Finally, with the inertial, gravity, aerostatic and control forces incorporated, Li establishes the dynamics model of a flexible airship as a single set of nonlinear differential equations, which can be linearized.

In this thesis, an assumption is made that the airship is a rigid-body vehicle. The airship's model is derived from the equations of motion for a rigid-body vehicle moving in vacuum, as described by Li [3]. Then the relevant solid-fluid interaction forces, both aerostatic and aerodynamic, are incorporated into the equations. This is exactly the same procedure that was followed by Li [3]. This nonlinear dynamics model can then be implemented in a dynamics simulation program to simulate the movements of the airship in wind.

The implementation of these techniques is described in detail in Chapter 2 of this thesis.

1.3 Thesis Objectives and Organization

1.3.1 Objectives

The overall purpose of this thesis is to design a control system for a tethered airship, to control the position of the airship in nominal wind conditions. The static and dynamic airship model parameters need to be calculated and modelled in a computer simulation. The effect of wind on the airship needs to be modelled in the simulation, as well as the effect the tether has on the airship. A microprocessor computer interface with a global positioning system (GPS), wind sensors and electrical actuators needs to be designed and built into the gondola of the airship. Finally the functionality of the controlled airship needs to be demonstrated during flight tests.

The purpose of the research and development that are done during this project is to study the possibility and feasibility of developing geostationary Stratolites.

1.3.2 Organization

Each chapter of this thesis is dedicated to elaborate on one of the main objectives. Chapter 1 gives an overview of the project and some background information on the current state of similar research projects. Chapter 2 focuses on the development of a dynamic model for an airship and the implementation of the simulation. Chapter 3 focuses on the design of the gondola and all the hardware interfaces. Chapter 4 elaborates on the different software that was written and how the different parts of the system are organised in a user friendly interface. Chapter 5 is dedicated to how the controllers work and how they were designed. Chapter 5 also shows the results obtained from flight tests, while Chapter 6 concludes the project and gives some recommendations for future projects.

1.4 On Notations

The author has tried to follow conventional notations for different physical variables. However, this may cause some confusion because of the differing conventions in different fields. For example, q denotes both pitch rate in flight dynamics and dynamic pressure in aerodynamics. The symbol for dynamic pressure was changed to q_0 for the duration of this thesis.

Chapter 2

Dynamic Modelling of an Airship

A number of airship models have been derived by various authors, as can be seen in the abstract of [4]. The model that was derived by Yuwen Li, in [4], was mainly used in this thesis to develop a non-linear airship model for simulation purposes. A few things differ from the model that was developed by Yuwen Li: This model, for instance, has to incorporate the effect of a tether. The effects of control surface deflections were neglected, due to the fact that the airship that was used in this thesis does not have a controllable elevator or rudder.

The modelling begins by deriving the equations of motion for an airship in a vacuum. This is done in Paragraph 2.1. The interaction forces and moments between the airship and air are then derived. The derivations of the aerostatic and aerodynamic characteristics are formulated in Paragraph 2.2 and Paragraph 2.3 respectively. The final model is simulated in MATLAB, by using Simulink. The simulation is described in Paragraph 2.4.

2.1 Equations of Motion for Airship in Vacuum

The equations of motion for a 6-DOF vehicle are usually derived in the body axis. The body axis is indicated by Figure 2.1 as $\{oxyz\}$, while the inertial axis system is indicated as $\{OX_IY_IZ_I\}$. The position of the airship can be described by a vector written in inertial coordinates, $[x_I, y_I, z_I]^T$, while the orientation of the airship is represented by the Euler angles: (roll, pitch and yaw), $[\phi, \theta, \psi]^T$.

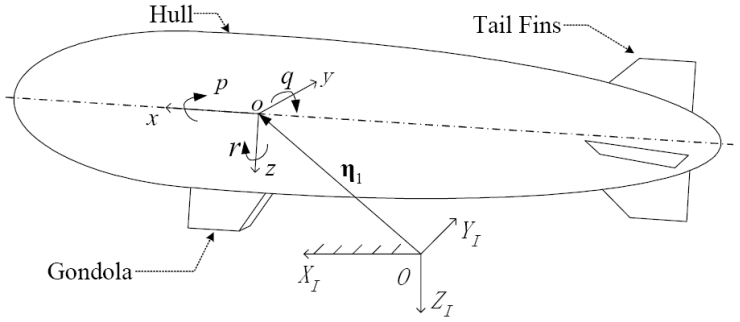


Figure 2.1: Body axis and Inertial axis system (Source: [3])

The 6-DOF equations for a vehicle moving in a vacuum are summarised as:

$$\boldsymbol{M_{rigid}}\dot{\mathbf{V}} = \boldsymbol{\tau_I} + \boldsymbol{\tau_G} + \boldsymbol{\tau_C} \tag{2.1}$$

$\mathbf{V} = [\mathbf{v}^T, \boldsymbol{\omega}^T]^T$; where $\mathbf{v} = [u, v, w]^T$ denotes the translational velocity vector and $\boldsymbol{\omega} = [p, q, r]^T$ denotes the angular velocity vector. Both these vectors are expressed in the body axis. $\boldsymbol{\tau_I}$, $\boldsymbol{\tau_G}$ and $\boldsymbol{\tau_C}$ denote the terms from inertia, gravity and control respectively. These terms are described in a later paragraph.

$\boldsymbol{M_{rigid}}$ is the mass matrix of the rigid airship expressed in Equation 2.2.

$$\boldsymbol{M_{rigid}} = \begin{bmatrix} m\boldsymbol{I_{3\times 3}} & -m\boldsymbol{r_g^\times} \\ m\boldsymbol{r_g^\times} & \mathbf{J} \end{bmatrix} \tag{2.2}$$

The m in Equation 2.2 is the total mass of the airship. This mass includes the masses of the hull, gas, fins, gondola and the tether. The respective masses of each part of the airship that was used in this project is summarised in Table 2.1.

$m_{hull} = 5.7kg$
$m_{fins} = 1.4kg$
$m_{H_2} = 1.2kg$
$m_{gondola} = 3.5kg$
$m_{tether} = 1.0kg$
$m = 12.8kg$

Table 2.1: Mass properties of airship

The **J** in Equation 2.2, is the inertia tensor of the airship. The assumption was made that the airship rotates aerodynamically around its centre of volume (COV). The inertia tensors of the gondola and the fins around their own centres of mass (COM) were ignored. The inertia tensor is also known as the second moment of inertia and is calculated as follows:

$$J_{xx} = \frac{1}{20}\rho_{H_2}(V_H)D^2 + m_{gondola}r_{gc}^2 + m_{fins}r_{fc}^2 + m_{hull}r_{hc}^2 \tag{2.3}$$

$$J_{yy} = \frac{1}{20}\rho_{H_2}(V_H)(L^2 + D^2) + m_{gondola}r_{gc}^2 + m_{fins}r_{fv}^2 + m_{hull}r_{hv}^2 \tag{2.4}$$

$$J_{zz} = \frac{1}{20}\rho_{H_2}(V_H)(L^2 + D^2) + m_{fins}r_{fv}^2 + m_{hull}r_{hv}^2 \tag{2.5}$$

Table 2.2 gives the values and explanations of the various airship dimensions that were used in Equations 2.3 to 2.5. The value of ρ_{H_2} is given in Table 3.1. Figure 2.2 shows the airship dimensions graphically.

Equations 2.3, 2.4 and 2.5 can be solved to give the following inertia tensor matrix:

$$\mathbf{J} = \begin{bmatrix} 7.81 & 0 & 0 \\ 0 & 50.0 & 0 \\ 0 & 0 & 46.5 \end{bmatrix} \tag{2.6}$$

The $\boldsymbol{r_g}$ vector is basically the distance from the centre of volume to the centre of gravity of the airship. The centre of gravity of the airship will be underneath the centre of volume due to the

Volume of the airship (V_H)	$14.11m^3$
Length of the airship (L)	$8.0m$
Diameter of the airship (D)	$1.9m$
Average radius from hull to body x-axis (r_{hc})	$0.75m$
Average radius from hull to body y-axis and z-axis (r_{hv})	$2.0m$
Distance from gondola to body axis origin (r_{gc})	$1.0m$
Distance from COV to COM (r_g)	$0.9m$
Distance from COM of fins to body x-axis (r_{fc})	$0.8m$
Distance from COM of fins to body y-axis and z-axis (r_{fv})	$3.75m$

Table 2.2: Airship dimensions with explanations

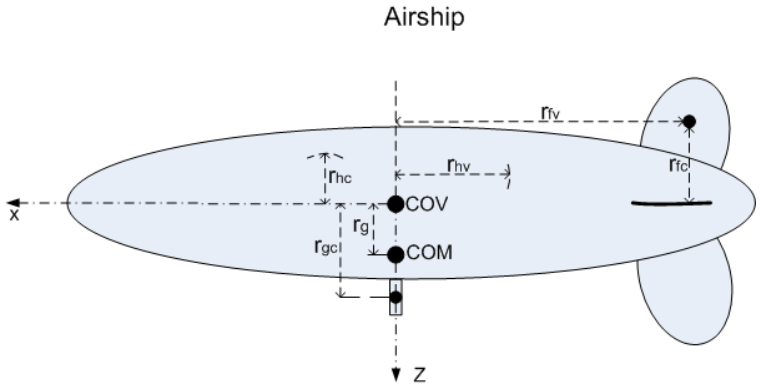


Figure 2.2: Airship dimensions

weight of the gondola. The skew symmetric matrix, $\boldsymbol{r_g^\times}$, corresponding to this vector is given in Equation 2.7.

$$\boldsymbol{r_g^\times} = \begin{bmatrix} 0 & -0.9 & 0 \\ 0.9 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{2.7}$$

Equation 2.7 and 2.6 can now be used to compute the mass matrix given in Equation 2.2. The mass matrix is as follows:

$$\boldsymbol{M_{rigid}} = \begin{bmatrix} 12.8 & 0 & 0 & 0 & 11.5 & 0 \\ 0 & 12.8 & 0 & -11.5 & 0 & 0 \\ 0 & 0 & 12.8 & 0 & 0 & 0 \\ 0 & -11.5 & 0 & 7.81 & 0 & 0 \\ 11.5 & 0 & 0 & 0 & 50.0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 46.5 \end{bmatrix} \tag{2.8}$$

The right hand side of Equation 2.1 consists of external forces and moments. These forces and moments denote the terms from inertia, gravity and control respectively, and can be described as follow:

The inertial force and moment is calculated as:

$$\boldsymbol{\tau_I} = \begin{bmatrix} -m\boldsymbol{\omega}^{\times}\mathbf{v} + m\boldsymbol{\omega}^{\times}\boldsymbol{r_g}^{\times}\boldsymbol{\omega} \\ -m\boldsymbol{r_g}^{\times}\boldsymbol{\omega}^{\times}\mathbf{v} - \boldsymbol{\omega}^{\times}\mathbf{J}\boldsymbol{\omega} \end{bmatrix} \tag{2.9}$$

The gravitational force ($\boldsymbol{F_G}$) and moment is calculated as:

$$\boldsymbol{\tau_G} = \begin{bmatrix} \boldsymbol{F_G} \\ \boldsymbol{r_g}^{\times}\boldsymbol{F_G} \end{bmatrix} \tag{2.10}$$

The gravitational force is given as:

$$\boldsymbol{F_G} = mg \times \begin{bmatrix} -\sin\theta \\ \cos\theta\sin\phi \\ \cos\theta\cos\phi \end{bmatrix} \tag{2.11}$$

The control forces and moments are due to the thrusts of the two motors. Each motor can be controlled manually or by the use of the automatic control system. Figure 2.3 shows a graph of the approximate amount of thrust, in Newton (N), which each motor can supply. This graph was obtained through practically measuring the thrust of a motor for different motor speeds. The thrusts were measured in a wind-free environment.

A moment can be applied to the airship by applying a positive thrust to one of the motors and a negative thrust to the other motor. The efficiency of the propellers vary depending on the direction in which the motor turns. A maximum forward thrust is approximately $2N$ while a maximum reverse thrust can only generate a force of approximately $1.5N$. This means that the efficiency of a propeller turning backwards is 75% of the forward efficiency.

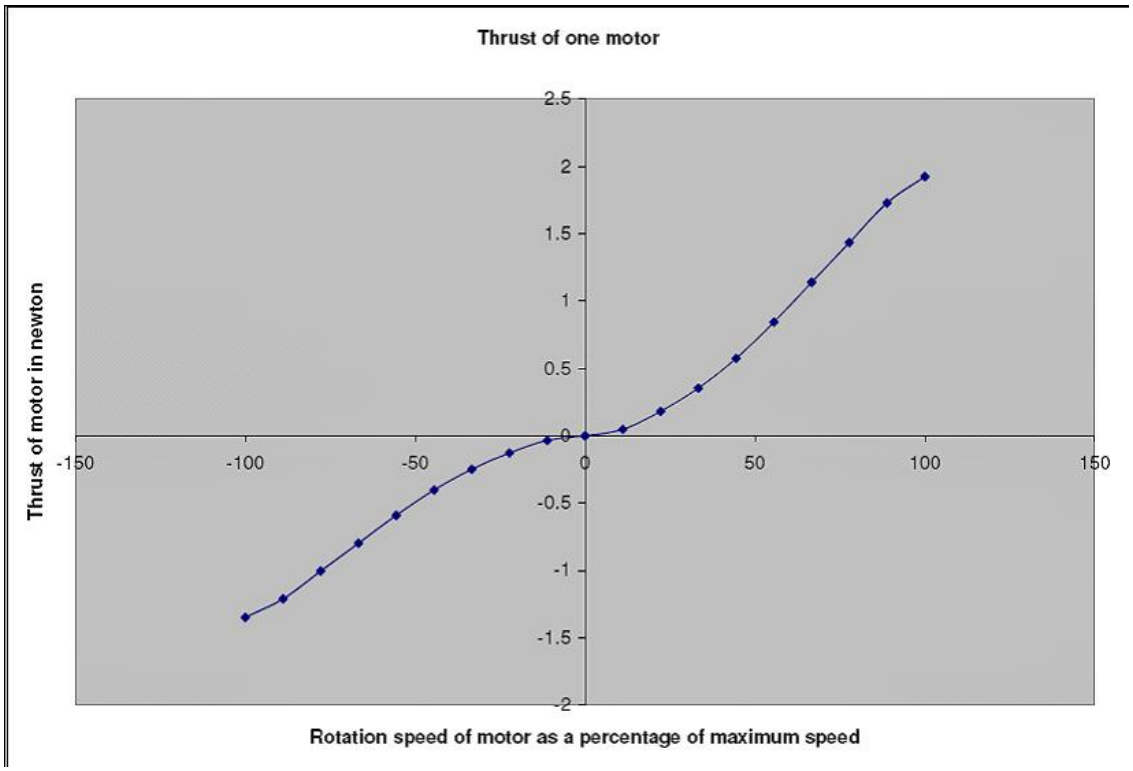


Figure 2.3: Thrust of each motor in Newton

Equation 2.12 shows the forces and moments created by the motors. This is called the control equation.

$$\boldsymbol{\tau_C} = \begin{bmatrix} T_{motor1} + T_{motor2} \\ 0 \\ 0 \\ - - - - - - - - - - \\ 0 \\ 0.65 \times (T_{motor1} + T_{motor2}) \\ 0.6 \times (T_{motor1} - T_{motor2}) \end{bmatrix} \tag{2.12}$$

The values 0.65 and 0.6 in Equation 2.12 are the distances in meters between the motors and the relevant axis.

2.2 Aerostatics

The aerostatic force and moment refers to the buoyancy of the airship. This relates to the static air pressure surrounding the airship and is independent of the motion of the airship. The aerostatic moment is zero because the body frame was chosen at the centre of volume of the airship. Equation 2.13 shows the aerostatic force and moment as a vector.

$$\boldsymbol{\tau_{AS}} = \rho_{air} g V_H \begin{bmatrix} \sin\theta \\ -\cos\theta \sin\phi \\ -\cos\theta \cos\phi \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{2.13}$$

V_H is the volume of the airship and ρ_{air} is the air density. The aerostatics equation is incorporated into the equations of motion by adding it to the right hand side of Equation 2.1.

2.3 Aerodynamics

Unlike the aerostatic force and moment which is independent of motion, the aerodynamic forces and moments are dependent on the motion of the airship. The aerodynamic forces are categorised in various terms based on the physical effects of that force. This chapter explains the aerodynamic effects of the added-mass force and moment (Paragraph 2.3.1), the viscous effect (Paragraph 2.3.3), the axial drag (Paragraph 2.3.5), side force (Paragraph 2.3.6), lift (Paragraph 2.3.7), the effect of the fins (Paragraph 2.3.4) and the effect of the tether (Paragraph 2.3.8).

2.3.1 Added-mass force and moment

The first effect that is included is called the added-mass force and moment. The added-mass can be described briefly as an apparent mass that is due to the axial forces acting on the hull of the airship due to an acceleration of the airship. The equations for estimating the added-mass force

and moment are given in Equations 2.14 and 2.15. Equation 2.14 is a 6×6 symmetric matrix. The elements of the 3×3 matrices, $M_{11}, M_{12}, M_{21}, M_{22}$, are calculated in Paragraph 2.3.2.

$$\boldsymbol{M_A} = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \tag{2.14}$$

$$\boldsymbol{\tau_A} = -\begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{v}} \\ \dot{\boldsymbol{\omega}} \end{bmatrix} - \begin{bmatrix} \boldsymbol{\omega}^{\times}(M_{11}\mathbf{v} + M_{12}\boldsymbol{\omega}) \\ \boldsymbol{v}^{\times}(M_{11}\mathbf{v} + M_{12}\boldsymbol{\omega}) + \boldsymbol{\omega}^{\times}(M_{21}\mathbf{v} + M_{22}\boldsymbol{\omega}) \end{bmatrix} \tag{2.15}$$

The first term in Equation 2.15 relates to the time rates of change of the linear and angular velocities. The second term relates to the coupling of the linear and angular velocities. To incorporate these terms into the dynamic model, the first term is written on the left hand side of Equation 2.1 so that the mass matrix $\mathbf{M}_{rigid}$ is replaced by $\mathbf{M}_{rigid} + \mathbf{M}_A$; while the second term is added to the right hand side of Equation 2.1. This term is also referred to as $\boldsymbol{\tau_M}$ for use in Equation 2.66.

2.3.2 Estimation for the Added-mass matrix

This chapter describes how the elements of the 3×3 matrices, $M_{11}, M_{12}, M_{21}, M_{22}$, are calculated. Both the hull and the fins affect the added-mass matrix. The effects of the hull and fins are calculated separately.

If the origin of the body axis is situated at the COV, which is the case in this project, then all the off-diagonal terms in the added-mass matrix that are due to the hull are zero.

Equation 2.16 shows the effected terms due to the hull.

$$\boldsymbol{M_{AH}} = \begin{bmatrix} m_{H,11} & 0 & 0 & 0 & 0 & 0 \\ 0 & m_{H,22} & 0 & 0 & 0 & 0 \\ 0 & 0 & m_{H,33} & 0 & 0 & 0 \\ 0 & 0 & 0 & m_{H,44} & 0 & 0 \\ 0 & 0 & 0 & 0 & m_{H,55} & 0 \\ 0 & 0 & 0 & 0 & 0 & m_{H,66} \end{bmatrix} \tag{2.16}$$

Equations 2.17 to 2.20 show how to solve this part of the added-mass matrix. The k_1, k_2 and k' factors, that are used in Equations 2.17 to 2.20, were acquired from Figure 2.4, for a fineness ratio of 4.2. The fineness ratio is easily acquired by dividing the length of the airship by the diameter of the airship.

$$m_{H,11} = k_1 \rho_{air}(V_H) = 1.38kg \tag{2.17}$$

$$m_{H,22} = m_{H,33} = k_2 \rho_{air}(V_H) = 15.04kg \tag{2.18}$$

$$m_{H,44} = \frac{1}{20} k' \rho_{air}(V_H) D^2 = 2.03kgm^2 \tag{2.19}$$

$$m_{H,55} = m_{H,66} = \frac{1}{20} k' \rho_{air}(V_H)(L^2 + D^2) = 37.98kgm^2 \tag{2.20}$$

The added mass and moment of inertia due to the fins can be computed by solving Equations 2.21 to 2.24. x_{fs} and x_{fe} are the x-coordinates of the start and end positions of the fins. These

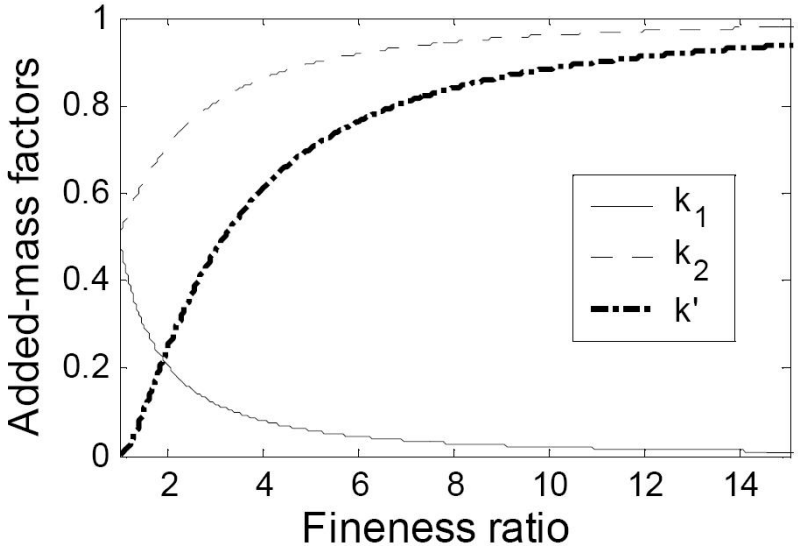


Figure 2.4: Inertia factors (Source: [4])

coordinates were measured and turned out to be $3.3m$ and $3.9m$ respectively. η_f is an efficiency factor of 0.35. This factor was obtained from [4].

$$m_{f,22} = m_{f,33} = \eta_f \int_{x_{fs}}^{x_{fe}} m_{s,22} dx = 0.71kg \tag{2.21}$$

$$m_{f,35} = -m_{f,26} = -\eta_f \int_{x_{fs}}^{x_{fe}} m_{s,22} x dx = -2.54kgm \tag{2.22}$$

$$m_{f,44} = \eta_f \int_{x_{fs}}^{x_{fe}} m_{s,44} dx = 0.24kgm^2 \tag{2.23}$$

$$m_{f,55} = m_{f,66} = \eta_f \int_{x_{fs}}^{x_{fe}} m_{s,22} x^2 dx = 9.17kgm^2 \tag{2.24}$$

The added-mass distribution constants of the fins, $m_{s,22}$ and $m_{s,44}$, are calculated in Equations 2.25 and 2.26 respectively. R in these equations is the body cross-sectional radius of $0.5m$ and b is the fin semi span which is $1.1m$. k_{44} was obtained from [4] and can be approximated to have a value of 1.

$$m_{s,22} = \rho_{air}\pi\Big(b - \frac{R^2}{b}\Big)^2 = 2.93kg/m \tag{2.25}$$

$$m_{s,44} = \frac{2}{\pi} k_{44} \rho_{air} b^4 = 1.14kgm \tag{2.26}$$

The added-mass matrix of the fins is now given in Equation 2.27.

$$\boldsymbol{M_{AF}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & m_{f,22} & 0 & 0 & 0 & m_{f,26} \\ 0 & 0 & m_{f,33} & 0 & m_{f,35} & 0 \\ 0 & 0 & 0 & m_{f,44} & 0 & 0 \\ 0 & 0 & m_{f,35} & 0 & m_{f,55} & 0 \\ 0 & m_{f,26} & 0 & 0 & 0 & m_{f,66} \end{bmatrix} \tag{2.27}$$

The total added-mass matrix can be calculated by adding the two matrices of Equation 2.16 and Equation 2.27. The result of the added-mass matrix is shown in Equation 2.28.

$$\boldsymbol{M_A} = \left[\begin{array}{ccc|ccc} 1.38 & 0 & 0 & 0 & 0 & 0 \\ 0 & 15.75 & 0 & 0 & 0 & 2.54 \\ 0 & 0 & 15.75 & 0 & -2.54 & 0 \\ \hline 0 & 0 & 0 & 2.27 & 0 & 0 \\ 0 & 0 & 0 & 0 & 47.15 & 0 \\ 0 & 0 & 0 & 0 & 0 & 47.15 \end{array}\right] = \left[\begin{array}{c|c} M_{11} & M_{12} \\ \hline M_{21} & M_{22} \end{array}\right] \tag{2.28}$$

2.3.3 Viscous effect on the hull

The force normal to the centerline due to viscous effects (F_{NV}) can be computed as given in Equation 2.29. This equation was acquired from [4].

$$F_{NV} = -q_0 \sin(2\gamma)(k_2 - k_1) \int_{\epsilon_0}^{L} \left(\frac{ds}{d\epsilon}\right) d\epsilon + q_0 \eta C_{DC} \sin^2 \gamma \int_{\epsilon_0}^{L} 2R d\epsilon \tag{2.29}$$

The term q_0 refers to the dynamic pressure of the air surrounding the airship and is calculated as:

$$q_0 = \frac{1}{2} \rho_{air} \times |v_0|^2 \tag{2.30}$$

γ is the angle between the centerline and the velocity vector. Equation 2.31 shows how this angle is calculated.

$$\gamma = \tan^{-1}\left(\frac{\sqrt{v^2 + w^2}}{u}\right) \tag{2.31}$$

k_1 and k_2 are the same factors that was used previously and was acquired from Figure 2.4.

ϵ denotes the longitudinal position from the nose of the airship and ϵ_0 denotes the location at which the flow of air ceases to be potential. ϵ_0 is calculated as:

$$\epsilon_0 = (0.378 \times L) + (0.527 \times \epsilon_1) = 6.45m \tag{2.32}$$

ϵ_1 denotes the position along the x-axis where the area of the hull decreases most rapidly and is at $6.5m$ for the specific hull.

R and S is the cross-sectional radius and area of the airship at the specific position ϵ.

The integrals of Equation 2.29, can now be solved by integrating from ϵ_0 to the tail of the airship at $L = 8.0m$. The integrals are approximated by subdividing the hull into intervals which are $0.5m$ apart and calculating the area or radius of each interval, before adding these values to obtain the solution of the integral. Equations 2.33 and 2.34 show the solutions of these integrals.

$$\int_{\epsilon_0}^{L} (\frac{ds}{d\epsilon}) d\epsilon = \frac{0.452 + 0.396 + 0.283}{0.5} \times 0.5 = 1.131 \tag{2.33}$$

$$\int_{\epsilon_0}^{L} 2R d\epsilon = (1.065 + 0.765 + 0.3) \times 0.5 = 1.065 \tag{2.34}$$

η is an efficiency factor accounting for the finite length of the body and is determined from the fineness ratio of the body. Reference [4] gives this factor to be $\eta = 0.62$.

C_{DC} is the cross-flow drag coefficient of an infinite-length circular cylinder. In [22] this coefficient is approximated as a constant of 1.2.

Equation 2.29 can now be simplified as follows:

$$F_{NV} = (-q_0 \sin(2\gamma) \times 0.89) + (q_0 \sin^2 \gamma \times 1.45) \tag{2.35}$$

The force due to viscous effects can now be written in vector form as follows:

$$\boldsymbol{F_v} = -F_{NV} \times \begin{bmatrix} 0 \\ \frac{-v}{\sqrt{v^2+w^2}} \\ \frac{-w}{\sqrt{v^2+w^2}} \end{bmatrix} \tag{2.36}$$

The moment due to the viscous effect can be computed by using:

$$M_v = -q_0 \sin(2\gamma)(k_2 - k_1) \int_{\epsilon_0}^{L} \Big(\frac{ds}{d\epsilon}\Big)(\epsilon_m - \epsilon) d\epsilon + q_0 \eta C_{DC} \sin^2 \gamma \int_{\epsilon_0}^{L} 2R(\epsilon_m - \epsilon) d\epsilon \tag{2.37}$$

ϵ_m is the distance from the origin of the body frame to the nose of the airship. This distance is $3.5m$. The moment due to the viscous effect can be simplified to give the following equation:

$$M_v = -q_0 sin(2\gamma) \times (1.642) + q_0 sin^2 \gamma \times (5.085) \tag{2.38}$$

The moments can be written in vector form as follows:

$$\boldsymbol{M_v} = M_v \times \begin{bmatrix} 0 \\ \frac{w}{\sqrt{v^2+w^2}} \\ \frac{-v}{\sqrt{v^2+w^2}} \end{bmatrix} \tag{2.39}$$

The viscous forces and moments, as described in Equation 2.40, can now be added to the right hand side of Equation 2.1, to incorporate these forces and moments into the dynamic model of the airship.

$$\boldsymbol{\tau_V} = \begin{bmatrix} \boldsymbol{F_v} \\ \boldsymbol{M_v} \end{bmatrix} \tag{2.40}$$

2.3.4 Force and moment acting on the fins

The next step is to calculate the force acting on the fins, normal to the centerline of the airship. This is obtained by estimating the force distribution and integrating this over the fin area. The equations for doing this were developed in [4]. These equations can be applied as follows:

The forces acting on the fins can be separated into body axis components. This results in Equations 2.41 to 2.43. The forces are due to the velocity at which the fins are moving through the air. The only forces, acting on the fins, that has an effect on the airship is the forces normal to the centerline of the airship.

$$F_x = 0 \tag{2.41}$$

$$F_y = F_n \times \left(\tan^{-1}\left(\frac{v_{nzp}}{u}\right) + \tan^{-1}\left(\frac{v_{nzn}}{u}\right)\right) \tag{2.42}$$

$$F_z = F_n \times \left(\tan^{-1}\left(\frac{v_{nyp}}{u}\right) + \tan^{-1}\left(\frac{v_{nyn}}{u}\right)\right) \tag{2.43}$$

F_n is described in Equation 2.44.

$$F_n = q_0 S_F \Delta C_{p\alpha} \times \frac{180}{\pi} \tag{2.44}$$

F_n is a function of the dynamic pressure, as was the case for the viscous effect described in Paragraph 2.3.3. S_F is the area of one fin, which is equal to $0.32m^2$. $\Delta C_{p\alpha}$ is the pressure coefficient of the fins. In [8] this coefficient is given as 0.05 per degree.

v_{nzp}, v_{nzn}, v_{nyp} and v_{nyn} describes the relative velocity of the fins through the air. Equations 2.45 to 2.48 describe these relative velocities mathematically.

$$v_{nzp} = -v_w + (r_{fc} \times p) + (r_{fv} \times r) \tag{2.45}$$
$$v_{nzn} = -v_w - (r_{fc} \times p) + (r_{fv} \times r) \tag{2.46}$$
$$v_{nyp} = -w_w - (r_{fc} \times p) - (r_{fv} \times q) \tag{2.47}$$
$$v_{nyn} = -w_w + (r_{fc} \times p) - (r_{fv} \times q) \tag{2.48}$$

The terms v_w and w_w are the airship velocity relative to the wind.

Figure 2.5 graphically shows how these relative velocities are interpreted.

The moments acting on the fins are calculated in Equations 2.49 to 2.51.

$$M_x = r_{fc} \times F_n \times \left(\tan^{-1}\left(\frac{v_{nyp}}{u}\right) - \tan^{-1}\left(\frac{v_{nzp}}{u}\right) - \tan^{-1}\left(\frac{v_{nyn}}{u}\right) + \tan^{-1}\left(\frac{v_{nzn}}{u}\right)\right) \tag{2.49}$$

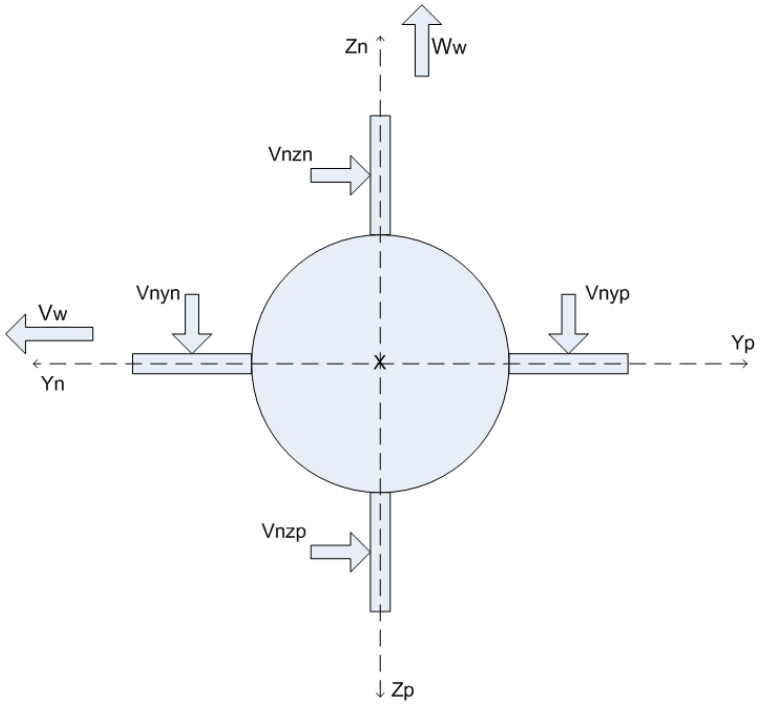


Figure 2.5: Forces acting on fins

$$M_y = r_{fv} \times F_y \tag{2.50}$$

$$M_z = -r_{fv} \times F_z \tag{2.51}$$

The forces and moments acting on the fins, as combined in Equation 2.52 can now be added to the right hand side of Equation 2.1, to incorporate these forces and moments into the dynamic model of the airship.

$$\boldsymbol{\tau_F} = \begin{bmatrix} F_x \\ F_y \\ F_z \\ M_x \\ M_y \\ M_z \end{bmatrix} \tag{2.52}$$

2.3.5 Axial Drag

Axial drag is a force that opposes the forward movement of the airship. The magnitude of this force depends on the surface area of the airship, the velocity of the airship and the angle of attack of the airship. Drag coefficients must be determined through experiments or tests, as was done on the YEZ-2A airship by Gomes in [23]. For the airship used in this project, the drag coefficient was derived as follows:

$$C_D \simeq \frac{0.05}{\cos^2(2\alpha)}, \quad for \quad \alpha \leq 30^\circ \tag{2.53}$$

$$C_D \simeq 0.23 \times \cos\alpha, \quad for \quad \alpha > 30^\circ \tag{2.54}$$

α is the angle of attack given as:

$$\alpha = \tan^{-1}\left(\frac{w}{u}\right) \tag{2.55}$$

The equation for calculating the drag force is given as:

$$F_{drag} = -q_0 C_D S_H \tag{2.56}$$

q_0 is the dynamic pressure as given in Equation 2.30. S_H is the surface area of the airship. This area is given as:

$$S_H = V_H^{\frac{2}{3}} = 5.84m^2 \tag{2.57}$$

The axial drag force is along the body x-axis of the airship. The side force described in Paragraph 2.3.6 and the lift described in Paragraph 2.3.7 are combined with the axial drag to give the opposing forces in all body axes. This can be seen in Equation 2.64.

2.3.6 Side force

The side force is a force that opposes any sideways motion of the airship, in the same way as the drag force opposes the forward movement of the airship. This side force depends on the sideslip angle (β), calculated as:

$$\beta = \tan^{-1}\left(\frac{v}{u}\right) \tag{2.58}$$

The side force varies according to the dynamic pressure, side force coefficient and surface area. It is described as:

$$F_{sideforce} = -q_0 C_Y S_H \tag{2.59}$$

The side force coefficient can be approximated as follows, due to data extracted from [23]:

$$C_Y \simeq 1.2 \sin\beta \tag{2.60}$$

2.3.7 Opposing lift force

The opposing lift force originates from the same principles as the drag and side force. This force opposes the vertical movement of the airship. In this project the vertical movement will be minimal due to the tether which keeps the airship at a relatively constant altitude, and therefore limits the airship's vertical movement considerably. Equation 2.61 formulates this force:

$$F_{lift} = -q_0 C_L S_H \tag{2.61}$$

In this case q_0 is still the dynamic pressure and S_H is still the same surface area as before. The

lift coefficient was extracted from tests that were done in [23]. C_L is given as a function of α_l in Equation 2.63, where α_l is given as:

$$\alpha_l = \tan^{-1}\left(\frac{v}{w}\right) \tag{2.62}$$

$$C_L \simeq \sin\alpha_l \tag{2.63}$$

The axial drag, side force and opposing lift can now be combined as a vector to give the following force to be added to the equations of motion of an airship. This is done by adding Equation 2.64 to the right hand side of Equation 2.1.

$$\boldsymbol{\tau_D} = \begin{bmatrix} F_{drag} \\ F_{sideforce} \\ F_{lift} \end{bmatrix} \tag{2.64}$$

2.3.8 Effect of the tether

The effect of the tether must be added to the dynamic model of the airship. Practically the tether will limit the altitude of the airship. If it is assumed that the airship is anchored by the tether at the origin of the inertial axis system and the distance, η_1 in Figure 2.1, can not exceed the length of the tether. The distance from the origin of the inertial axis system to the airship is calculated as:

$$\eta_1 = \sqrt{x_I^2 + y_I^2 + z_I^2} \tag{2.65}$$

Figure 2.6 graphically illustrates what is meant by the distance η_1.

When this distance is equal to the length of the tether, the force by which the tether will pull the airship will be equal to all the forces acting on the airship. This will allow the airship to be in a state of equilibrium.

It is also assumed that the force in the tether will be zero whenever the distance from the origin of the inertial axis system to the airship is smaller than the length of the tether. In this situation the airship is able to move freely as if no tether is attached to it.

The sum of all the forces acting on the airship must be calculated in inertial coordinates in order to know the magnitude of the force in the tether. This is done by adding all the forces on the right hand side of Equation 2.1 together and transforming the resultant force into inertial coordinates. This results in:

$$\boldsymbol{F_{total}} = \mathbf{A}^T \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \left(\boldsymbol{\tau_I} + \boldsymbol{\tau_G} + \boldsymbol{\tau_C} + \boldsymbol{\tau_{AS}} + \boldsymbol{\tau_M} + \boldsymbol{\tau_V} + \boldsymbol{\tau_F} + \boldsymbol{\tau_D}\right) \tag{2.66}$$

$\mathbf{A}^T$ is the transformation matrix that is used to transform body coordinates to inertial coordinates. $\mathbf{A}$ is given as:

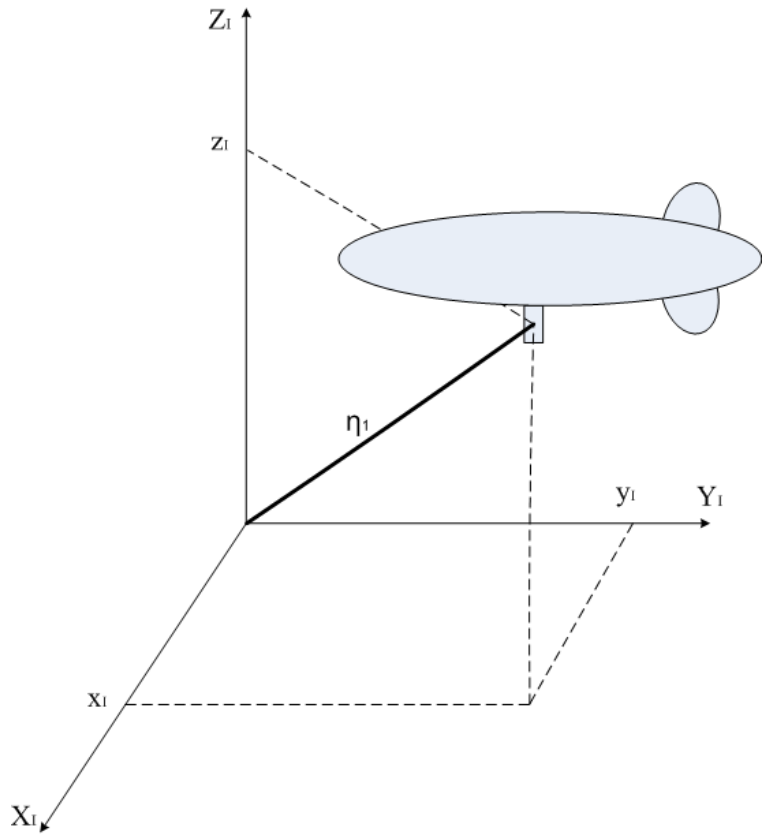


Figure 2.6: Airship with tether

$$\mathbf{A} = \begin{bmatrix} \cos\psi\cos\theta & \sin\psi\cos\theta & -\sin\theta \\ (\cos\psi\sin\theta\sin\phi) - (\sin\psi\cos\theta) & (\sin\psi\sin\theta\sin\phi) + (\cos\psi\cos\phi) & \cos\theta\sin\phi \\ (\cos\psi\sin\theta\cos\phi) + (\sin\psi\sin\phi) & (\sin\psi\sin\theta\cos\phi) - (\cos\psi\sin\phi) & \cos\theta\cos\phi \end{bmatrix} \tag{2.67}$$

Equation 2.68 shows the force in the tether that cancels the total force acting on the airship, in order to maintain a state of equilibrium. This force exists in the tether whenever $\eta_1 \approx L_{tether}$.

$$\boldsymbol{F_{tether_I}} = \frac{-(|\boldsymbol{F_{total}}|)}{\eta_1} \times \begin{bmatrix} x_I \\ y_I \\ z_I \end{bmatrix} \tag{2.68}$$

η_1 should not be able to exceed the length of the tether, therefore the force in the tether must increase dramatically whenever η_1 tends to exceed the length of the tether in an effort to pull the airship back.

Equation 2.69 shows the force in the tether that pulls the airship back whenever η_1 exceeds the length of the tether. The factor, 1000, is the elasticity coefficient of the tether which prevents the tether from stretching when the tether is at its maximum length.

$$\boldsymbol{F_{tether_I}} = \frac{-(|\boldsymbol{F_{total}}| + (1000 \times (\eta_1 - L_{tether})))}{\eta_1} \times \begin{bmatrix} x_I \\ y_I \\ z_I \end{bmatrix} \tag{2.69}$$

Now that the force in the tether has been calculated in inertial coordinates, the force in the tether should be transformed back to body coordinates and added to the right hand side of Equation 2.1 to incorporate this force in the equations of motion of the airship. Equation 2.70 shows how the transformation is done, while Equation 2.71 gives the total effect of the tether as a matrix.

$$\boldsymbol{F_{tether_B}} = \mathbf{A}\boldsymbol{F_{tether_I}} \tag{2.70}$$

$$\boldsymbol{\tau_T} = \begin{bmatrix} \boldsymbol{F_{tether_B}} \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{2.71}$$

2.4 Simulation

Now that the equations of motion for an airship have been derived, it is necessary to create a simulation to see whether the model that has been obtained has the expected behaviour, under various conditions.

Simulink is used to create a block diagram model for this simulation. A MATLAB s-function is written to incorporate the equations of motion that was derived in the previous sections. Paragraph 2.4.1 elaborates on the development of the s-function. Figure 2.7 shows the Simulink block diagram.

After the simulation is created, it is of utmost importance to compare the results of various simulated conditions with practical measurements of the actual airship behaviour. These results are shown and are explained in Paragraph 2.4.2.

2.4.1 S-Function

The first step of developing the s-function is to identify the relevant input and output parameters that the simulation should have: The input variables should include all the external parameters that can change the behaviour of the airship. These include the actuators that will be used to control the airship, the length of the tether and the wind direction and wind velocity. The most important output parameters are those that are needed to control the airship, namely the position of the airship and the heading of the airship. Other output parameters like the roll, and pitch angles can also be monitored as well as the velocity of the airship and the angular rates at which the airship turns.

The main purpose of the s-function is to keep track of all the states of the airship, and to update the state vector at regular intervals. The state vector has an initial value at the start of the simulation. This state vector will be updated according to all the forces and moments that are applied to the airship at a specific sample rate.

The constant parameters that define the airship, like the mass of the airship, are all included in the s-function. The parameters that depend on the state of other parameters, like the added-mass of the airship, are also included according to the equations that were defined in the previous sections of Chapter 2.

Figure 2.7 shows how the Simulink block diagram links the input and output parameters to the s-function.

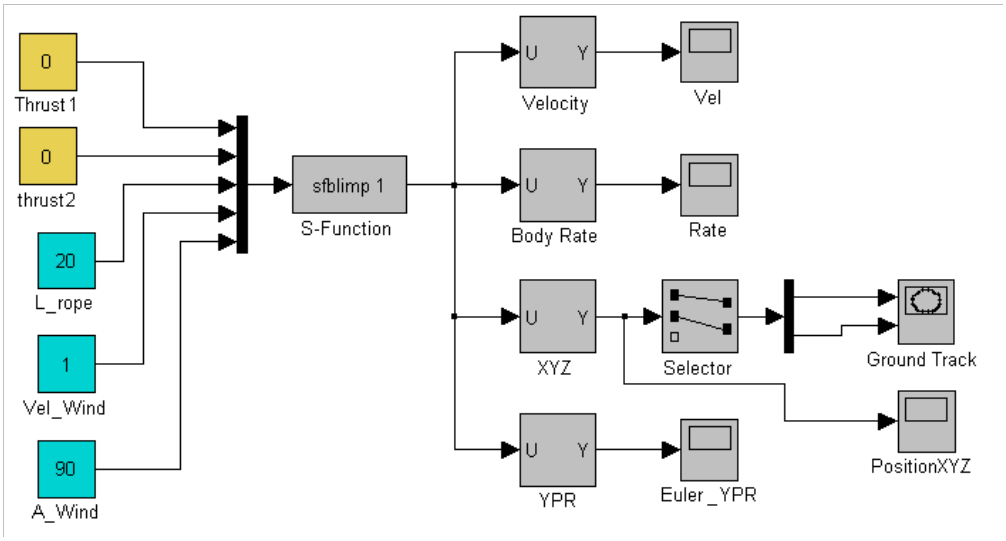


Figure 2.7: Simulink Block Diagram

2.4.2 Results

The accuracy of the model can be verified by comparing the outputs of the simulation with the measurements taken during flight tests. Having an accurate model of the airship has great advantages when it comes to designing a control system for the airship: It allows the engineer to test the controllers before any actual flights are attempted. It also allows the engineer to simulate various conditions in an attempt to predict the limitations of the system.

This Paragraph shows the outcome of a few simulations to illustrate how the airship is expected to react in the presence of wind. Paragraph 5.1.1 and Paragraph 5.2.1 are dedicated to comparing the simulation with actual flight test data in an effort to identify a linear system for use during control system design.

Simulation 1

The first simulation illustrates the movement of the airship in a constant wind of $2m/s$. The direction of this wind is initially perpendicular to the airship. The length of the tether was set to $20m$ in this case.

Figure 2.8 shows the damped yawing movement of the airship. It clearly shows the effect that the fins have in turning the airship into the direction of the wind.

Figure 2.9 shows the altitude, z_I, of the airship in inertial coordinates. This figure clearly shows how the airship lifts from the ground, and how the altitude is eventually limited by the tether.

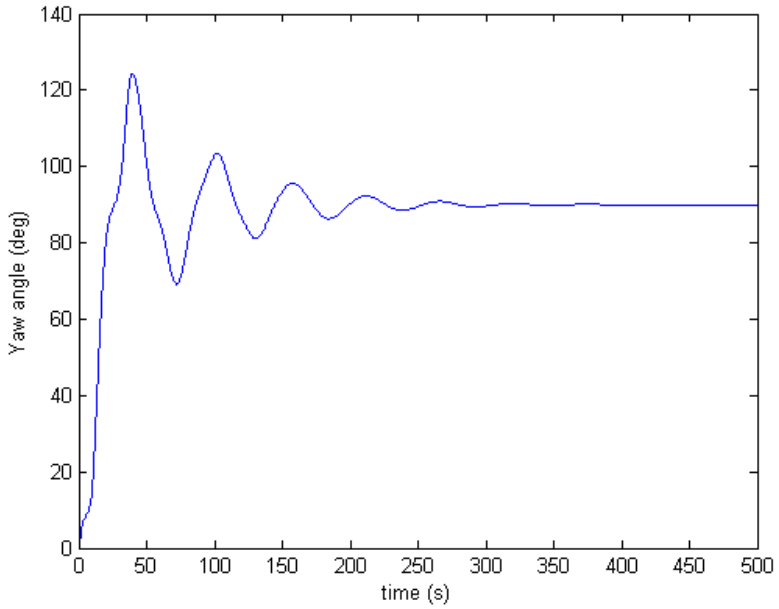


Figure 2.8: Simulation result: Yaw angle

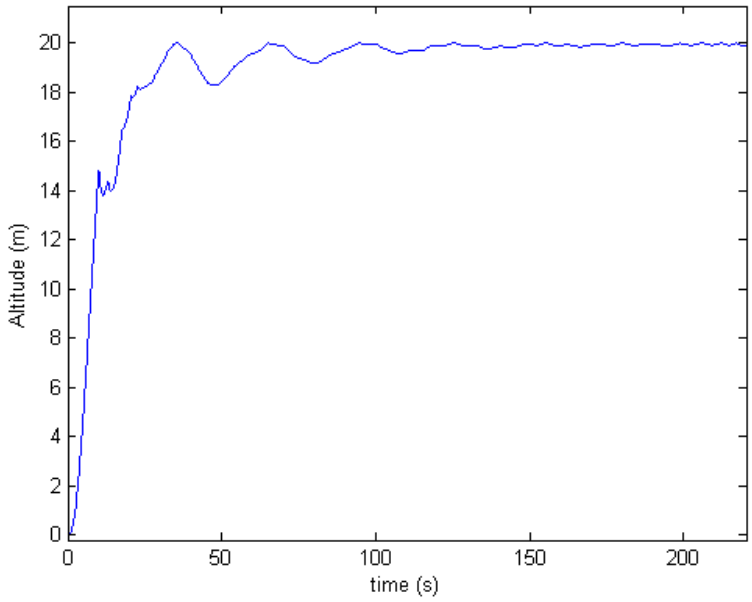
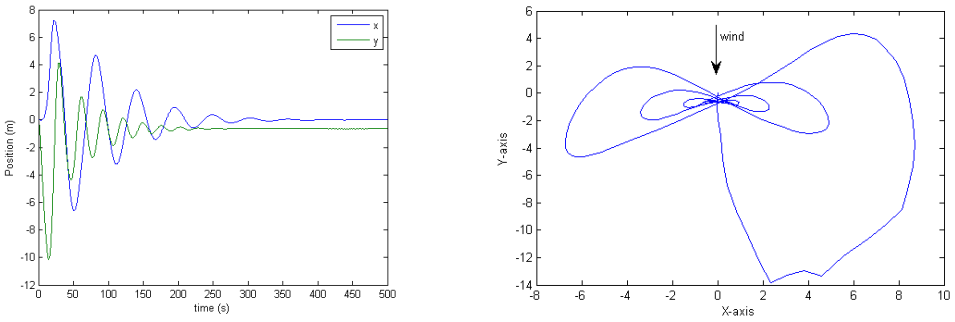


Figure 2.9: Simulation result: Altitude

Figure 2.10 shows the x_I and y_I positions of the airship. This figure shows how the airship settles at a position away from the origin, which is in the opposite direction to the wind. The distance that the airship is away from the origin depends on the wind velocity.



(a) Position over time (b) $X_I Y_I$ plot

Figure 2.10: Simulation result: $X_I Y_I$ Position

Simulation 2

The second simulation illustrates how the x_I position of the airship changes for different wind velocities. The angle of the wind is directly from the front of the airship in all the following cases. Therefore the y_I position will be zero. The length of the tether is still $20m$ as in the previous simulation.

Figure 2.11 shows the position of the airship when a constant wind step input of $1m/s$ was applied.

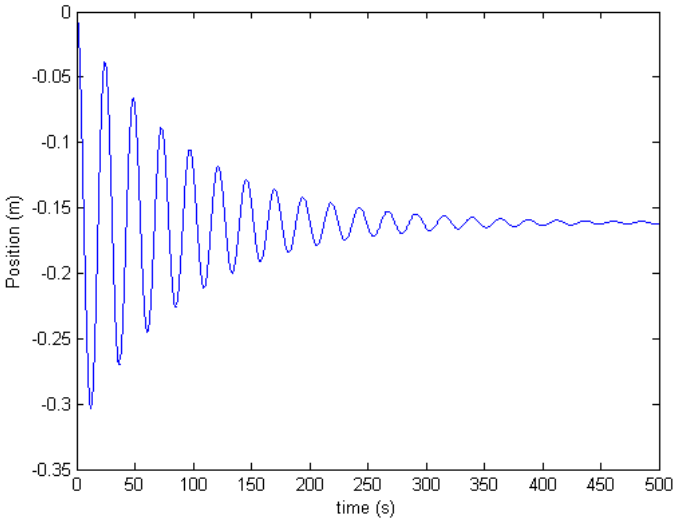


Figure 2.11: Simulation result: X_I Position (wind = 1 m/s)

Figure 2.12 shows the position of the airship when a constant wind step input of $2.5m/s$ was applied.

Figure 2.13 shows the position of the airship when a constant wind step input of $4m/s$ was applied.

The increase in the offset position from the origin can be noted in the respective figures. The oscillation period is the same in all cases, because the length of the tether didn't change. The

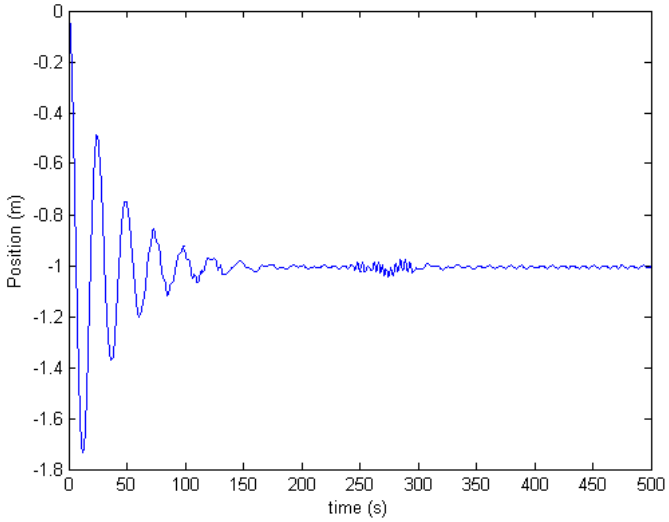


Figure 2.12: Simulation result: X_I Position (wind = 2.5 m/s)

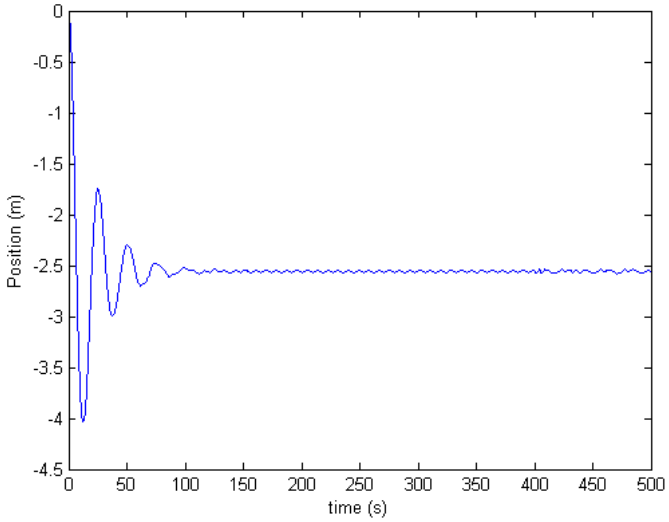


Figure 2.13: Simulation result: X_I Position (wind = 4 m/s)

damping of the oscillations increases dramatically when the velocity of the wind increases, because of the increase in the drag force acting on the airship.

Simulation 3

The third simulation shows the effects that the actuators have on the airship. Figure 2.14 shows the result of applying a continuous yaw-moment of $1.8Nm$ to the airship. No wind is present during this simulation.

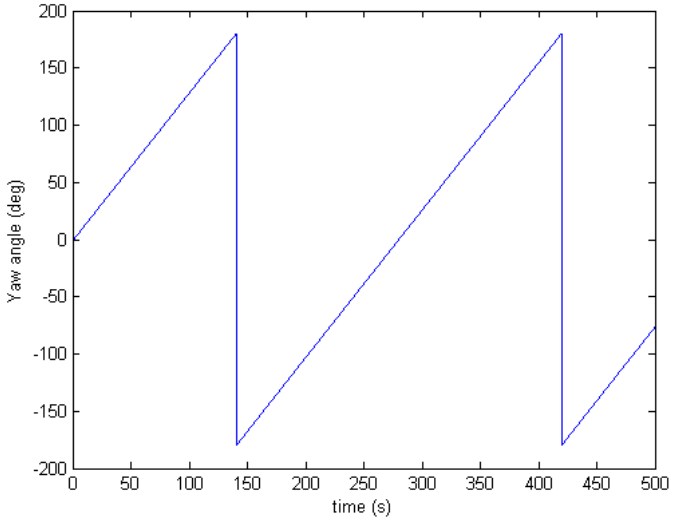


Figure 2.14: Simulation result: Yaw angle due to applied moment

According to this simulation it will take the airship $280s$ to rotate through 360° with the applied yaw-moment. This is extremely slow. The rotation rate can be increased by moving the motors further away from each other, or by using stronger motors with bigger propellers. Extra actuators can also be used to increase the available actuating power.

Figure 2.15 shows the result of applying a forward thrust of $2N$ by each actuator. There is no wind present during this simulation.

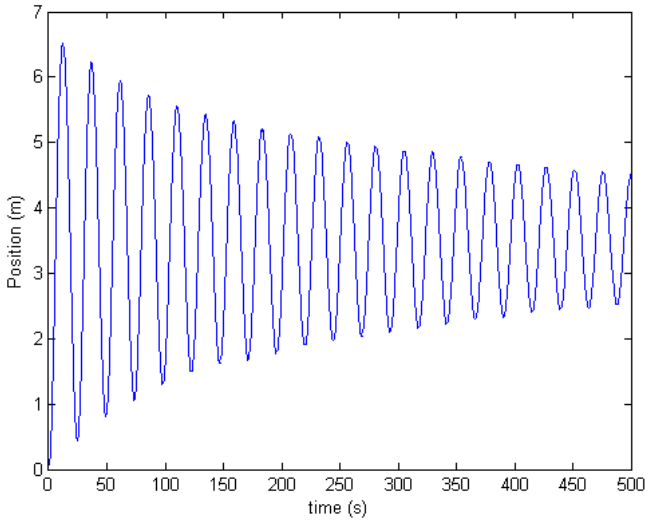


Figure 2.15: Simulation result: X_I Position due to applied thrust (0 m/s wind)

The damping of the oscillations is very low when no wind is present, but will increase when wind is present, as shown in Figure 2.16. The offset by which the applied thrust can move the airship is between 3 and 4 meters when using a $20m$ tether.

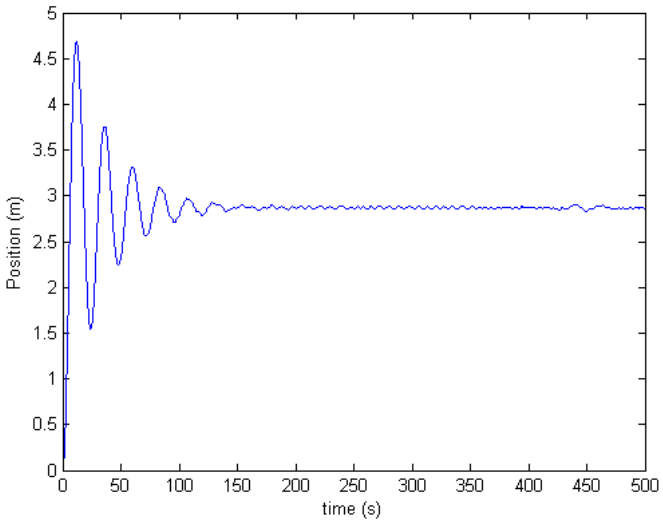


Figure 2.16: Simulation result: X_I Position due to applied thrust (2 m/s wind)

The control system which is implemented in Paragraph 5.2 will have to apply the correct amount of forward thrust so that the position will settle at the origin of the inertial axis system.

Chapter 3

Hardware Design

Chapter 1 explained the need and use for designing and controlling a tethered airship. Chapter 2 explained how a tethered airship can be modelled for simulation purposes. In this chapter the focus shifts to the design and implementation of a physical airship system that can be used to control a tethered airship.

This airship system needs to be designed and built practically in order to make a study of the functionality and feasibility of such a controlled airship. Each section of the airship system is described in this chapter, together with an explanation of some of the basic principles concerning airships. This is needed in order to gain some understanding and insight on how the various systems of the airship should be designed to make it easier to implement a control system later on.

The airship system is divided into the following sections:

- The Hull, which is the body of the airship.
- The Gondola, which is fitted to the hull at the bottom of the airship.
- The Camera Tracker, which is situated on the ground and keeps track of the position of the airship.
- The Ground Station, to where all the data is communicated and from where the airship will be controlled.

These four sections of the airship system are all equally important. Without the hull, the airship system would not be able to fly. Without the gondola the airship system would not be able to be controlled. Without the ground station all the data that is measured by the camera tracker and the sensors on the gondola would be useless. But together, these four systems can be used to effectively control the position and the heading of the airship. This chapter does not focus on how the airship is controlled, but explains the design and purpose of each section. The controllers are explained in Chapter 5.

3.1 The Hull

The hull is also known as the body of the airship. The Gondola is fitted to the bottom of the hull while the four fins are fitted to the tail of the hull. The hull is basically a large container which has

a specific shape and size and can be filled with a LTA gas. The LTA gas gives volume to the hull of the airship and determines the amount of lift that the airship possesses. The shape of the airship with its body axis system is shown in Figure 3.1.

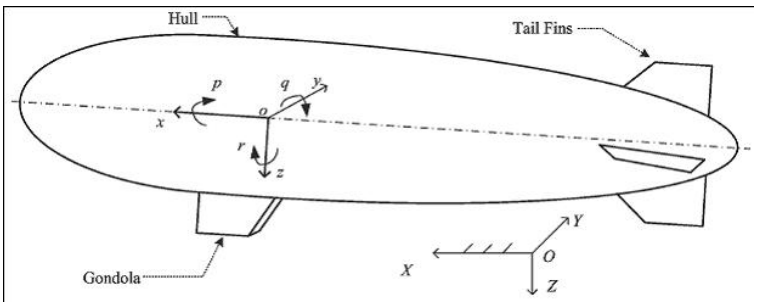


Figure 3.1: Body axis of airship (Source: [3])

Helium or hydrogen could be used to fill the hull, as both of these gases are LTA gases. There are some design trade offs which influence the decision of which gas to use. For example, hydrogen is a flammable gas, and although helium is not flammable, it is much more expensive than hydrogen. Hydrogen has half the density of helium and is capable of lifting a bigger mass with the same volume of gas.

The hull of the airship used in this project has a length (L) of 8 m, a diameter (D) of $1.9m$ and a volume (V_H) of $14.11m^3$. The ability of the airship to lift a mass depends on the density of the air surrounding the hull and the density of the LTA gas inside the hull. This is described as the principles of aerostatics in [8]. Aerostatics refers to the static buoyancy of any kind of body immersed in the atmosphere. The buoyancy force is equal to the weight of the air displaced by the body:

$$B = V_H \times \ \rho_{air} \tag{3.1}$$

Where:

- B is the upward buoyancy force acting on the body
- V_H is the volume of the hull
- ρ_{air} is the mean density of the air surrounding the airship

The buoyancy force acts on all bodies within the atmosphere but is usually very small when compared with the weight of the body. In the case of an airship the weight (W) can be made less than that of the displaced air, so that there will be a net upward lift (L_f) given by:

$$L_f = B \ - \ W \tag{3.2}$$

The weight (W) of the airship can be calculated as the weight of the hull, fins and the gas in the hull. W_0 refers to the weight of the hull and of the fins of the airship. This weight is $7.1kg$, as shown in Table 2.1.

$$W = V_H \times \ \rho_{gas} \ + W_0 \tag{3.3}$$

Equations 3.1, 3.2 and 3.3 can now be combined to give the excessive lift force (L_d), as seen in Equation 3.4. L_d is the lift force available for the payload. The payload consists mainly of the gondola but the weight of the tether also adds to the payload.

$$L_d = V_H \times (\rho_{air} - \rho_{gas}) \ - W_0 \tag{3.4}$$

The approximate densities of air, helium and hydrogen at sea level and 20℃, are given in table 3.1.

ρ_{air}	$1.225\ kg/m^3$
ρ_{He_2}	$0.169\ kg/m^3$
ρ_{H_2}	$0.084\ kg/m^3$

Table 3.1: Approximate densities of gases at sea level and 20℃

When substituting the values of Table 3.1 into Equation 3.4, the amount of excess lift could be calculated when using helium and hydrogen respectively.

$$L_{He_2} = \ 14.11 \ \times \ (1.225 - 0.169) - 7.1 = 7.80\ kg \tag{3.5}$$

$$L_{H_2} = \ 14.11 \ \times \ (1.225 - 0.084) - 7.1 = 9.0\ kg \tag{3.6}$$

The payload's weight should not exceed the excess lift. This is extremely important and has a major influence on the design of the gondola.

3.2 The Gondola

The gondola is the framework that contains all the electronic components, sensors and motors that are needed to control the airship.

The mass of the gondola needs to be less than the maximum amount of mass the airship can lift. This means that the gondola must weigh less than the mass calculated in Equation 3.5. Every design decision should take the weight restriction into account so that the gondola will not be too heavy in the end.

With this in mind, the decision was made to build the framework with PVC conduit. PVC conduit is strong but light and can be bought from any local hardware store. PVC conduit with a diameter of $25mm$ was mainly used to construct the gondola. Two pieces of PVC conduit with a diameter of $20mm$ were used to fit the gondola to the hull. Figure 3.2 shows the designed gondola from different angles.

Brushless DC motors were selected as actuators because they were relatively small for the amount of power they could deliver. The protective housing for the propellers was made from aluminium. Every electronic component which was used in the electronic circuit board, was carefully selected

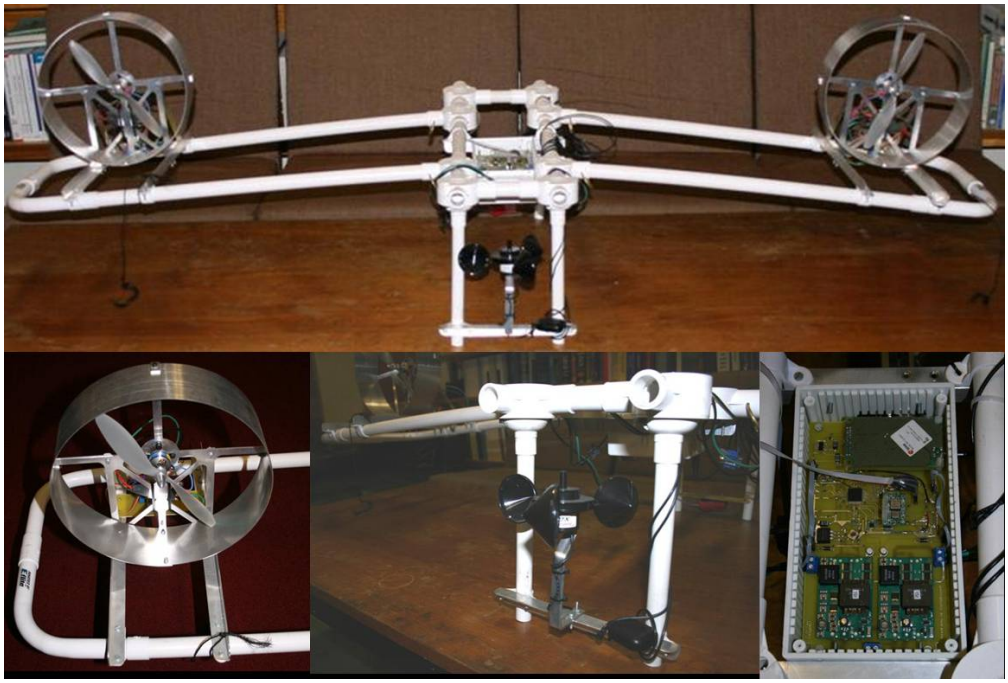


Figure 3.2: The gondola

so that the gondola would not exceed its weight restriction. In the end, the gondola weighed approximately $3.5kg$ which was well within the desirable limits.

The following electronic components were fitted into the electronic circuit board:

- Two isolated DC-DC converters to power the motors.
- A $5V$ DC regulator.
- A PIC microprocessor with quartz crystal clock.
- A GPS receiver module to measure position and velocity.
- A digital compass to measure the relative heading according to the magnetic field of the Earth.
- A two-axis accelerometer to measure the pitch- and roll angles.
- A gyroscope to measure yaw-rate.
- An electronic vane to measure wind direction.
- An anemometer to measure wind speed.
- A RS-232 serial communication interface.
- Various resistors, capacitors, transistors and diodes.
- A button to reset the PIC.
- Three LEDs.
- A $2.5V$ voltage reference.

- Two relays.
- Two electronic speed controllers (ESC's).

Figure 3.3 illustrates through a blockdiagram how all the components are connected to the PIC. The complete layout of the circuit board is included in Appendix A.

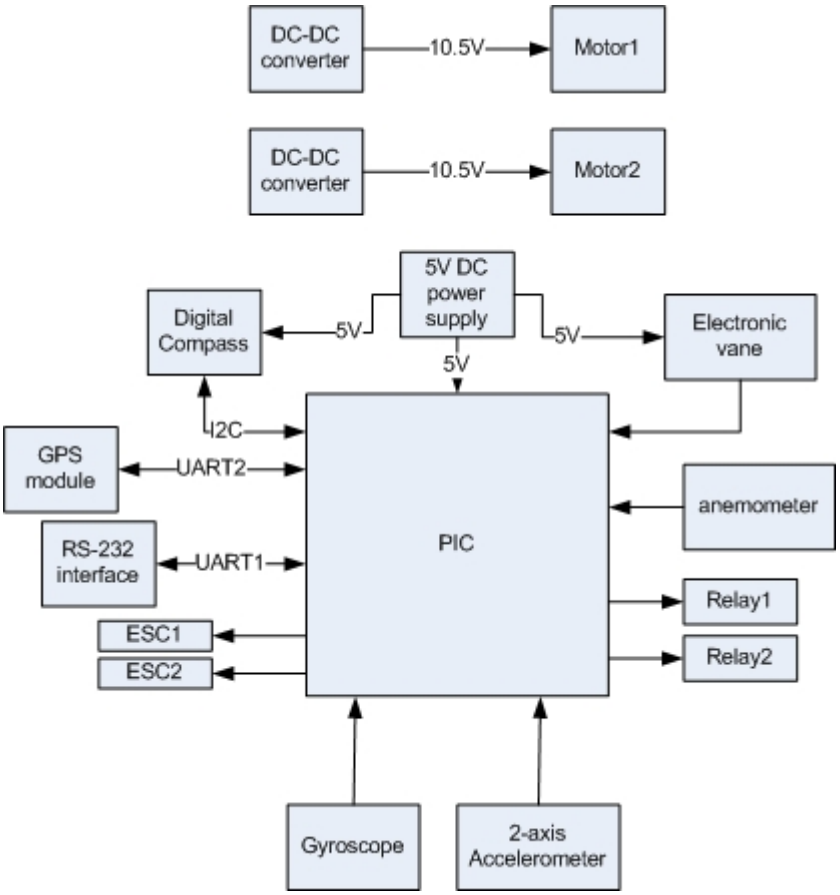


Figure 3.3: Blockdiagram of electronic circuit board

3.2.1 Micro-processor

The micro-processor that was used on the electronic circuit board is a high-performance, 16-bit, Digital Signal Controller, by Microchip. The dsPIC30f4011 [24] was used. The microprocessor is commonly referred to as the PIC. A quartz crystal with an oscillation frequency of $7.3728MHz$ was used to create the clock pulses at which the PIC execute commands. The electronic circuit board was also fitted with a reset button to restart the execution of the PIC software. The PIC software is described in Paragraph 4.1.

3.2.2 RS-232 Transceiver

Communication between the airship and a ground station was made possible through a wireless RS-232 transceiver. The transceiver was connected to the electronic circuit board through the RS-232 interface. The same RS-232 transceiver was connected to the ground station. The RS-232

transceiver transmits data at $115.2kbps$ over a frequency of $2.4GHz$. The features of the ground station are described in Paragraph 3.4.

3.2.3 Actuators

Two brushless DC motors were chosen to actuate the airship through two 12 inch propellers. The motors are driven with electronic speed controllers (ESCs) to control the speed of each motor. Each ESC receives pulses generated by the pulse width modulator (PWM) on the PIC. The width of these pulses varies between $1ms$ and $2ms$, which controls the motor's speed from zero to maximum speed respectively. Relays are used to switch the direction in which the motors turn. This is done by swapping two of the three cables that connect the ESC to the brushless DC motor. By doing this, the phase of the electric current that is applied to the motor changes direction which makes the motor turn in the opposite direction. The motors should be capable of creating a moment that can turn the airship around its z-axis. This is called a yawing moment. In order to generate a yaw moment, the motors should turn in opposite directions. The force generated by the thrust of the motors and the distance between the motors, influence the yawing moment according to Equation 3.7.

$$M_{yaw} = F_{thrust} \times d \tag{3.7}$$

The gondola was designed so that the distance (d) between the motors is $1.2m$. The motors are situated on the sides of the hull. The thrust generated by the motors is parallel to the hull's x-axis. This allows the motors to move the airship forward or backward along the x-axis when both motors turn with equal speeds in the same direction.

3.2.4 Power Configuration

The motors and circuit board are powered from the ground through a long cable, also known as the tether of the airship. In practise a high altitude platform would not be tethered, but batteries and solar panels could not be used to power this particular airship because of the weight restriction of the gondola. Because of energy losses in long cables, a DC voltage of $72V$ is carried through the tether towards the gondola. This high voltage is then converted by two respective DC-DC converters, to the proper voltage of $10.5V$, which is needed to power the two motors. Another small voltage regulator is used to obtain a $5V$ source to power the PIC and most of the other components on the electronic circuit board.

3.2.5 Wind Sensors

The two wind sensors are fitted to the gondola in such a way that no parts of the gondola obstruct the wind. This ensures that the accuracy of the wind measurements is as good as possible. The anemometer was calibrated by comparing the measurements of an already calibrated anemometer with the measurements of the used anemometer and adjusting the anemometer accordingly. Both the electronic vane and the anemometer plug into the electronic circuit board through $3.5mm$ stereo connectors. This allows these sensors to be easily disconnected from the electronic circuit board.

3.2.6 Digital Compass

The digital compass that was used is the *HMC6352* from Honeywell [5]. The digital compass is fitted as far away as possible from any current carrying conductors. The current inside a conductor creates a magnetic field around the conductor which will influence the measurement of the digital compass. Therefore, the digital compass is situated at the back of the gondola, away from any conducting metals. The digital compass connects to the electronic circuit board through a 4-wire ribbon cable.

The digital compass was tested to ensure that the measurements are not affected when the motors are switched on. It was seen that the motors don't interfere with the digital compass measurements. The digital compass has the ability to be calibrated to correct for hard-iron distortions of the earth's magnetic field. These distortions are due to magnetised materials in fixed locations near the digital compass. Therefore the digital compass are capable of making accurate heading measurements in most circumstances.

3.2.7 GPS Receiver Module

The GPS receiver module used is the *RCB-4H ANTARIS 4*, from U-Blox [25]. The GPS module is used mainly to measure the position of the airship. It can also be used to measure the velocity at which the airship is moving, the direction in which it is moving and the altitude of the airship. The problem with GPS measurements is that the normal GPS measurements are only accurate to approximately $4m$. This means that even though the airship is hanging perfectly still, the GPS measurements will show that the airship is actually moving inside a circle with a $4m$ radius. This lack in accuracy will not be a problem for a HAP flying at an altitude of $20km$, where this amount of variance in the position of the airship would have no remarkable effect on the station keeping of the HAP, but it will have a significant influence on controlling an airship at an altitude of just $20m$. In order to overcome this problem, the position of the airship needs to be measured in some other way, as described in Paragraph 3.3.

3.2.8 Gyroscope

The gyroscope is not actively used to control the airship. The gyroscope measures the yaw-rate of the airship and could have been used for controlling the heading of the airship, because yaw-rate is the first derivative of the yaw angle. The gyroscope is not used however, because the airship yaws very slowly and the noise of the gyroscope is substantial at low yaw-rates. The gyroscope is only used to monitor the yaw-rate. The $2.5V$ precision voltage reference is used by the gyroscope.

3.2.9 Accelerometer

The two-axis accelerometer is not used for control purposes either. An accelerometer can technically be used to control position, due to the fact that acceleration is the second derivative of position. It is however difficult to use the accelerometer for this purpose due to bias in the accelerometer output and large gravitational disturbances. In this case the accelerometer is only used to calculate the pitch and roll angles of the airship.

3.2.10 LED Configuration

There are three LEDs on the electronic circuit board. The one LED is on whenever the electronic circuit board is powered. The other two LEDs are connected to two of the output pins of the PIC and can have various functions. One LED is used to switch on whenever the GPS receiver has a satellite fix, for example. The use of these LEDs is defined in software.

3.3 The Camera Tracker

The Camera Tracker is a useful tool for measuring the position and velocity of the airship. The Camera Tracker consists of a camera that is at a fixed position on the ground. The camera is powered with a $9V$ power supply. The camera is connected through a parallel port to a laptop computer. The software on the laptop computer controls the camera. [1]

The aim of the camera software is to locate the brightest pixel in the field of view of the camera and to keep track of the coordinates of that specific pixel. The field of view of the camera can be estimated to have a square shape which consists of a thousand pixels by a thousand pixels, as seen in Figure 3.4. It is assumed that the pixel in the bottom-left corner is at the origin of an x, y axis system, so that every pixel can be described according to its x, y position. The pixel in the centre of the camera's field of view will have the coordinates of $500, 500$.

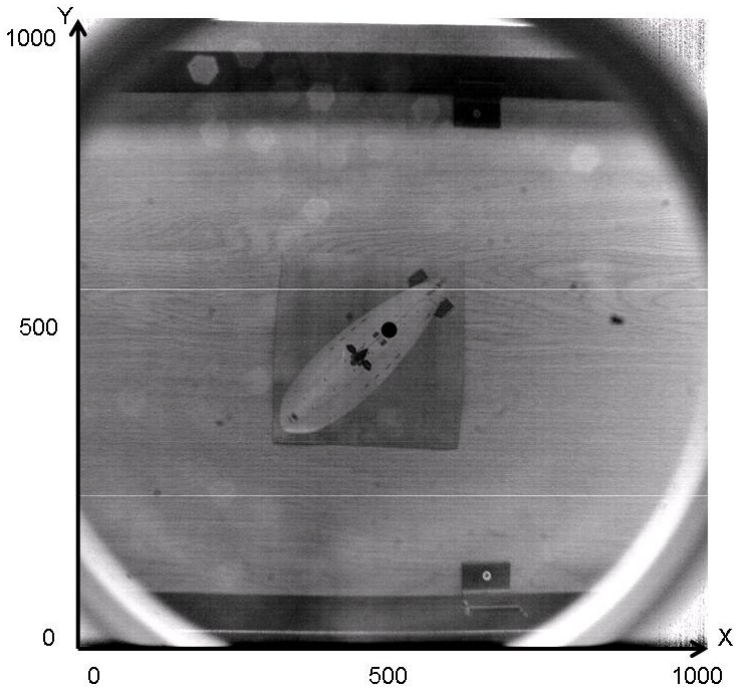


Figure 3.4: Camera Tracker's field of view

The position of the brightest pixel is sampled every $20ms$. The velocity of the pixel is determined by the software, through calculating the derivative of the position.

[1] The Camera Tracker was developed by Prof. W.H. Steyn

In order to determine the position of the airship, it is desirable to have an ultra-bright LED somewhere close to the centre of the airship. This ultra-bright LED will ensure that the brightest pixel in the field of view of the camera will always be the position of this LED. The Camera Tracker will locate the position of the airship and keep track of its movements by keeping track of the position of the ultra-bright LED.

The position coordinates of the ultra-bright LED are transmitted from the laptop computer through the serial port, to the Ground Station. The position of the airship can be calibrated to give the position in metric units. The scaling will depend on the altitude of the airship: The distance in meters that the airship moved will be drastically different if the airship is at an altitude of $5m$ and moves an amount of 50 pixels, than when the airship is at an altitude of $20m$ and moves the same 50 pixels. This calibration is done on the Ground Station and is described in Paragraph 4.2.

It is possible for the airship to move outside of the field of view of the camera. This will happen if the wind is too strong for the selected tether length. In such a case, the camera tracker will not be able to make accurate measurements of the airship's position. Figure 3.5 shows the maximum angle of the camera tracker's field of view.

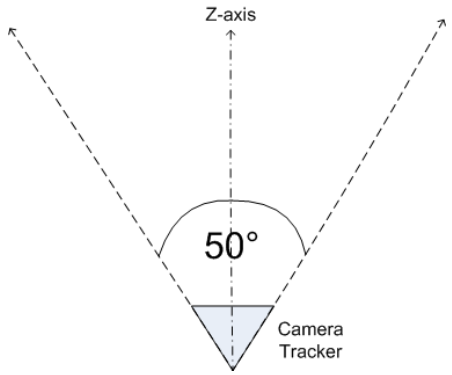


Figure 3.5: Maximum angle of Camera Tracker's field of view

The aperture of the camera can be changed manually to obtain the best accuracy in different light conditions.

3.4 The Ground Station

The Ground Station consists of a laptop computer with two serial ports. One of the serial ports is used to receive the positional data from the Camera Tracker through a serial cable. The RS-232 transceiver is connected to the other serial port. The transceiver is used to transmit and receive data between the airship and the Ground Station.

The following data packages are transmitted from the airship to the Ground Station:

- The GPS data, including time of week, latitude, longitude and height above sea level.
- The wind data, including wind speed and wind direction.
- The accelerometer data in both the x and y body axis.

- The gyroscope data which measures the yaw-rate of the airship.
- The digital compass data, which gives a relative heading according to the earth's magnetic field.
- The motor data, including the speed and direction of each motor.

All these data packages are received and stored in a binary file. The data can be extracted from the binary file so that various graphs can be plotted and analysed. This is done by the software described in Paragraph 4.2 and Paragraph 4.3 respectively.

The heading and the position of the airship can be controlled from the Ground Station, when the required control systems are activated. The appropriate control signals are then transmitted from the Ground Station to the airship to command the actuators and in return control the airship.

Other commands that can be sent from the Ground Station to the airship include a command to reset the PIC software, a command to reset the digital compass and various commands to manually control the actuators of the airship.

Chapter 4

Software Design

Software design is a fundamental aspect to successfully design an embedded system. The function of the software is to integrate all the different hardware components into one system that functions together. Every hardware subsystem should be programmed and thoroughly tested to work separately, before being incorporated into the full system. If any one of these hardware subsystems malfunctions due to a programming error, the whole system could be ineffective. This chapter elaborates on the necessary programming of each of the different subsystems and how each subsystem is incorporated into developing the final system.

The first part of this chapter describes the programming of the PIC. The PIC is the microprocessor on board the gondola of the airship. The different electronic sensors and the actuators, as described in Paragraph 3.2, are all integrated into a single system by use of the PIC.

The second part of this chapter describes the Ground Station software developed to monitor the sensor measurements and adjust controller parameters of the airship. The sensor data is displayed in real time on the Ground Station and saved to a binary file. These files can be extracted and the data can be plotted on graphs to be analysed. This software can be run on any personal computer or laptop running the Windows operating system.

The final part of this chapter describes the software developed in MATLAB to draw the graphs of all the relevant data saved by the Ground Station during test flights.

4.1 Software on the PIC

The PIC used, is the dsPIC30f4011 from Microchip. This PIC was chosen for the following features it possesses:

- Six PWM output channels: Two of which were used to control the speed of the motors.
- Supports I^2C^{TM} communication: Needed to interface with the digital compass.
- 16-bit Capture input functions: Needed to measure the pulses generated by the anemometer in order to calculate wind velocity.
- Timer modules: Three timers were used to synchronise some of the events executed by the PIC.

- 10-bit Analog to digital converter: Used to sample the voltage outputs of the accelerometer, gyroscope and wind vane.
- Two Universal Asynchronous Receiver or Transmitter (UART) modules: One UART was used to interface with the GPS module, and the other UART was used for the serial transceiver unit in order to communicate with the Ground Station.
- Multiple digital input and output pins: Used to switch certain circuits like the relays on or off.

Figure 4.1 shows the flow diagram of the PIC software's main function.

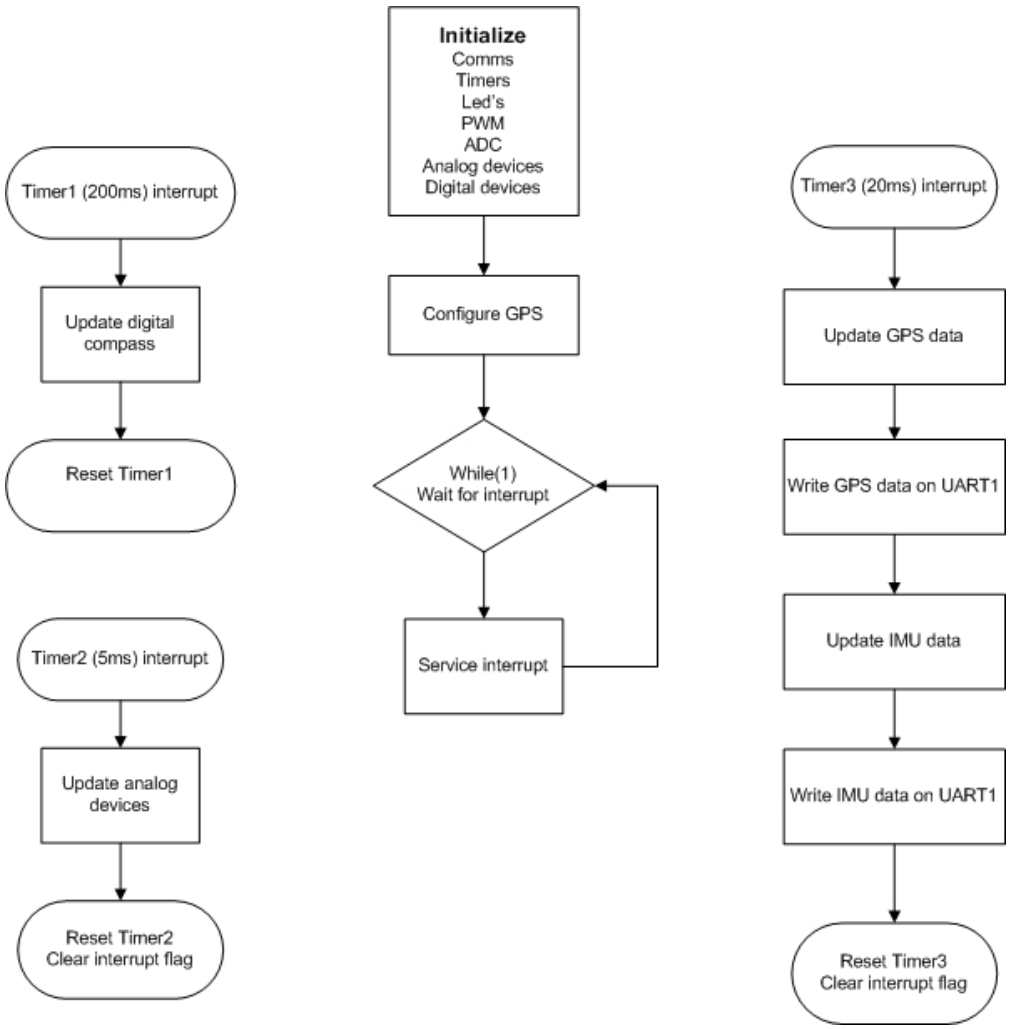


Figure 4.1: Flow diagram of PIC software's main function

The PIC software was coded in C. The MPLAB^{TM} IDE v8.02 software was used in conjunction with the C compiler to create a HEX file for programming the PIC. The PIC was programmed using a PICkit^{TM}2 programmer.

The following sections give a short overview on how each of the different modules on the PIC was set up, in an effort to integrate all the subsystems into a complete system.

4.1.1 The Timer modules

The use of Timers is essential for synchronizing different parts of the system. This system uses three different timers. Each timer is set up to have a specific overflow frequency. This means that every timer takes a specific time before it overflows. An interrupt procedure can be executed every time the timer overflows to execute a certain function. See Paragraph 4.1.2 for more information on interrupts.

The first thing that should be mentioned about the PIC is that the PIC executes commands at a certain frequency. The frequency at which a PIC executes commands depends on the oscillation frequency of the quartz crystal, which is $7.3728MHz$ in this case. This is called the clock frequency at which the system operates. This frequency can be divided into slower clock frequencies by factors of 2, 4, 8, 16, 32, 64, 128, 256 or 512. A timer consists of a 10-bit register. The value of the register is incremented by one at every clock cycle. When the 10-bit register reaches its maximum value of 1023 and when the register is incremented by one, the register overflows and the value becomes 0. At this point an interrupt could be requested. The rate at which these overflows occur can be set by changing the reload constant from where the register is incremented and by scaling the clock signal by one of the mentioned factors. By doing this, the timers can be set up to have overflow frequencies over a wide range of time, which can be extremely useful when used appropriately.

The first timer is set up to have an overflow frequency of $50Hz$. This correlates to a period of $20ms$ between consecutive timer register overflows and between consecutive interrupt requests. The interrupt procedure associated with this timer is the procedure that sends the most recent data, regarding the measurements of the respective sensors, to the Ground Station. This means that the measurements of the sensors on board the airship are updated every $20ms$.

This timer has another purpose as well: The digital compass is set up to take a new measurement once every $200ms$. This is exactly 10 intervals of the timer frequency, which means that a new measurement of the digital compass is taken at every tenth cycle of this timer. The new measurement will be repeated to the Ground Station during the other nine transmissions.

The second timer is a $5ms$ timer. This refers to a timer that executes a certain routine once every $5ms$ by the use of an interrupt. The routine associated with this interrupt takes measurements of the gyroscope and the two-axis accelerometer. The gyroscope and accelerometer are analog devices and the values of their outputs need to be converted into a digital format. This is done by the ADC module. See Paragraph 4.1.6 for details on the ADC module. The analog to digital conversions are completed every $5ms$ and the values are stored in buffers. The average of four consecutive analog to digital conversion values is calculated over $20ms$ to gain more accuracy in the measurement. These average values are the values sent to the Ground Station during the $20ms$ timer interrupt.

The last timer used is a 1 second timer. This timer is the slowest used in the system and is used in conjunction with the Input Capture module, described in Paragraph 4.1.4. Figure 4.2 shows the flow diagram of the Input Capture interrupt using this timer. The purpose of this timer is to measure the amount of time between two pulses of the anemometer. The period between two pulses is an indication of the wind velocity. The anemometer is calibrated to give the wind velocity in units of m/s. This is done by finding the period between two pulses, at a known independently measured velocity of the wind in m/s. For example, a calibration test for a wind of $1m/s$ will generate pulses at a time period of $0.85s$ as the anemometer turns. Table 4.1 shows the time period between pulses

for various wind velocities. It is shown in the flowdiagram of Figure 4.2 that the wind velocity is assumed to be zero whenever two consecutive pulses are more than one second apart.

Wind velocity	**Period of anemometer pulses**
$1m/s$	$0.85s$
$2m/s$	$0.68s$
$2.75m/s$	$0.55s$
$3m/s$	$0.51s$
$3.32m/s$	$0.45s$
$3.69m/s$	$0.39s$

Table 4.1: Wind velocity versus anemometer pulses

4.1.2 Interrupts

The use of interrupts is a crucial but delicate part of software development. Interrupts are a commonly used technique for computer multitasking, especially in real-time computing. The interrupt will essentially interrupt the procedure the processor is currently executing and immediately execute the interrupting procedure, before finishing the current procedure.

The PIC software makes use of eight interrupt procedures. Three of these interrupt procedures are servicing the three timers. These procedures are described in Paragraph 4.1.1.

Two of the interrupt procedures are servicing the UART communication. These two interrupt procedures are executed whenever data is received on the respective UART receiver inputs. The first UART module is used as a communicative link between the airship and the Ground Station. The commands sent from the Ground Station to the airship are received in the first UART module and an interrupt procedure is executed. The interrupt procedure interprets the command received by performing a switch statement. Each command starts with a specific 8-bit control character which selects a corresponding service to execute. Most of these commands involve the control of the motors. One command involves the resetting of the digital compass and another resets the PIC.

The second UART module is connected to the GPS receiver module. The data received by the GPS module is sent through this UART module to the PIC. The interrupt procedure accepts the data and runs a procedure to parse the received GPS data. This is necessary in order to obtain useful GPS data. The working of the GPS parser software is described in Paragraph 4.1.8.

The other interrupt procedures are the ADC interrupt, the PWM interrupt and the Input Capture interrupt.

The ADC interrupt is set up as described in Paragraph 4.1.6. This interrupt works in conjunction with the $5ms$ timer, as seen in Paragraph 4.1.1. After every set of analog to digital conversions, the specific ADC interrupt flag is set. This ADC interrupt flag must be cleared for the next conversion to take place.

The PWM module is set up as described in Paragraph 4.1.3. The interrupt procedure is executed whenever the PWM interrupt flag is set. The interrupt procedure clears the interrupt flag to generate the next PWM pulse.

The Input Capture module is set up as described in Paragraph 4.1.4. The Input Capture interrupt procedure is executed whenever there is a rise in the voltage on the Input Capture bus,

which connects the anemometer to the PIC. The time between consecutive interrupts indicates the speed at which the anemometer is turning, as an indication of the velocity of the wind. This interrupt procedure works in conjunction with the 1 second timer, as described in Paragraph 4.1.1.

Figure 4.2 shows the flow diagram of the Input Capture interrupt procedure.

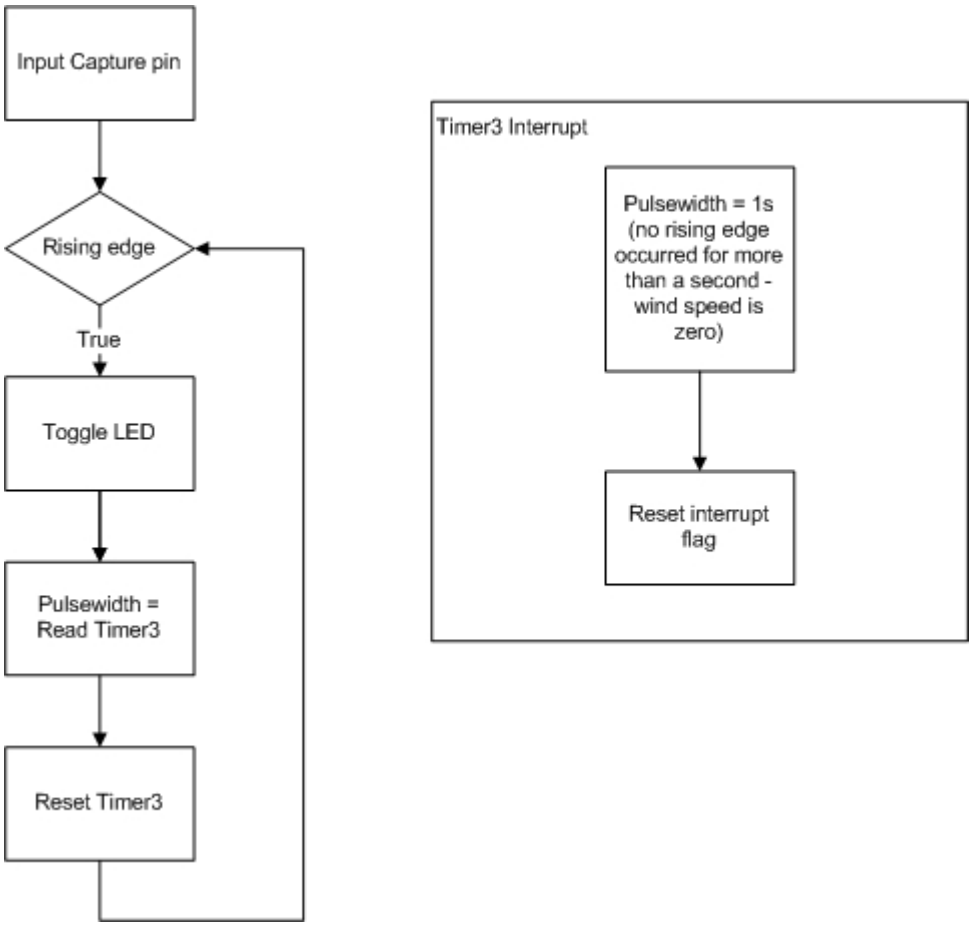


Figure 4.2: Flow diagram of Input Capture interrupt

Each interrupt has a specific priority allocated to it. This priority selects the order in which interrupt procedures should be executed. For example: When the processor is busy executing a specific procedure inside an interrupt of a specific priority, and another interrupt occurs, the processor will determine the priority of this new interrupt. If the priority of this interrupt is higher than the priority of the interrupt it was executing, the processor would interrupt the procedure it was executing and first service the interrupt procedure of higher priority. When the higher priority interrupt is finished, it will resume the interrupted procedure. Table 4.2 shows all the interrupts with its corresponding interrupt priorities. It shows the interrupts with the lowest priorities first. The interrupt with the highest priority is the Input Capture interrupt.

4.1.3 The PWM modules

The user manual, referred to in reference [26], specifies some macro functions to easily set up all the hardware modules of the PIC. These macros were used to set up the PWM module. The user

Interrupt name	Interrupt Priority
UART 1 RX interrupt	Priority 8
UART 2 RX interrupt	Priority 7
Timer 1 interrupt	Priority 6
Timer 2 interrupt	Priority 5
Timer 3 interrupt	Priority 4
ADC interrupt	Priority 3
PWM interrupt	Priority 2
Input Capture interrupt	Priority 1

Table 4.2: Interrupt priorities

specifies the values of the different parameters as defined by the macro function and the macro function in turn allocates these values to the correct hardware registers on the PIC. By setting the correct value to each parameter, the PWM module can have user specific characteristics.

In this case the PWM module was set up to have a frequency of $50Hz$. This is the specified frequency at which the electronic speed controllers of the motors operate. According to the datasheet of the PIC, referred to as [24], the following equation can be used to specify the required frequency:

$$T_{PWM} = \frac{T_{CY} \times (PTPER + 1)}{(PTMRPrescaleValue)} \tag{4.1}$$

By changing the duty cycle of this $50Hz$ signal, the speed of the motors can be varied. The airship has two motors to speed control separately. Therefore the need exists for two PWM output channels, one for each motor. The speed of the two motors can be adjusted independently by adjusting the duty cycle of each PWM output channel.

4.1.4 Input Capture module

The macros given in the PIC user manual [26] were used to set up the Input Capture module.

The parameters of the Input Capture module were set up to have the following characteristics: First of all the Input Capture module had to be allocated to work in conjunction with one of the timers in the timer module. The 1 second timer set up in Paragraph 4.1.1 was allocated for this purpose. Secondly the Input Capture module was instructed to request an interrupt on every rising edge of the Input Capture bus. The value of the 1 second timer will be stored during every interrupt, after which it will be cleared. The timer will then increment as usual, until the next interrupt occurs due to another rising edge.

4.1.5 The I^2C module

The macros in the PIC user manual [26] were used to set up the I^2C module.

The digital compass is designed to work with the I^2C communication protocol. The I^2C module of the PIC must be set up according to the specifications of the I^2C protocol of the digital compass. These specifications are listed in the datasheet of the digital compass, in [5]. In order to understand the set up procedure, it is necessary to first understand how the I^2C protocol works:

The I^2C protocol works at a specified baud rate. This baud rate is normally $100kbps$, according to the I^2C protocol standard. A whole group of I^2C devices can be connected to one system. In this

system however, there is only two I^2C devices. One of the devices in an I^2C system acts as a master device while the other devices are slave devices. Every slave device has its own unique 7-bit address. In this case the PIC is the master device and the digital compass is the slave device. The address of the digital compass is 42(hex). The master device sends 8-bit commands to the slave device and the slave device responds with an acknowledge bit with every command it receives. A clock signal with a frequency of $100kHz$ is used to synchronise the commands according to the specified baud rate. In order for two I^2C devices to communicate successfully, multiple commands must be sent and enough time must be allocated between commands, for the commands to be executed successfully, before another command can be sent. A start sequence is used to specify when a new command is about to be sent. These start sequences are generated by the PIC. After the command was sent and the execution of the command was fulfilled, a stop sequence is generated by the PIC. Figure 4.3 shows a graphical example of how I^2C communication works.

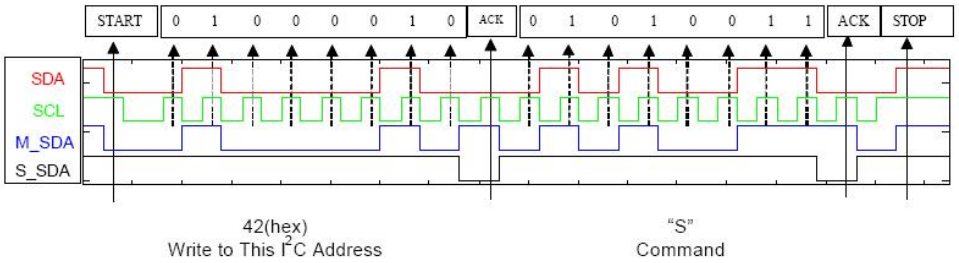


Figure 4.3: I^2C communication example (Source [5])

The datasheet of the digital compass, referred to in [5], gives a list of all the possible commands the digital compass will accept. The "Write to RAM Register" command is used during this project, to set up the digital compass correctly each time the system is started. The "Compensate and get new heading" command is used at specified intervals to retrieve the latest measurement data from the digital compass.

According to the datasheet of the digital compass [5], the heading measurements of the digital compass can be updated every $50ms$. The intervals at which the "Compensate and get new heading" command is used, are only once every $200ms$, and well within the required specification.

4.1.6 Analog to Digital Converter module

The Analog to Digital Converter (ADC) module is used to convert analog voltages to a 10-bit digital value. The ADC module uses reference voltages of $0V$ and $5V$. These voltages correspond to digital values of 0 and 1023 respectively. Any analog voltage between $0V$ and $5V$ is converted proportionally to a digital value between 0 and 1023 according to Equation 4.2.

$$Value = \frac{1023}{5} \times V_{analog} \tag{4.2}$$

The resolution of the ADC module is $4.88mV$. The resolution is calculated as:

$$Resolution = \frac{V_{CC}}{2^{10}} = \frac{5}{1024} \tag{4.3}$$

The ADC module was not set up using the macro functions of the PIC user manual [26]. Instead the appropriate values were directly assigned to the registers on the PIC.

First of all the inputs of the PIC must be defined as digital inputs or analog inputs. Only analog inputs can be converted to digital values with the ADC module. There are five analog inputs: They are connected to the gyroscope, the two axis of the accelerometer, the electronic wind direction vane and the temperature pin of the gyroscope.

The other registers that need to be set up is the ADCON1 and ADCON2 registers. These registers control the following attributes of the ADC module:

- The type of value the output values have. It is set to be unsigned integers.
- The timing device to use, is set to be the internal clock.
- The voltage references to use are the supply voltages powering the PIC.
- The number of conversions to do before an interrupt occurs. This is set to 5.
- The ADC module must be switched on.
- The ADC interrupt flag must be cleared to make sure that ADC interrupts will occur.

4.1.7 UART modules

The macros in the PIC user manual [26] were used to set up both the UART modules.

The first UART module is used by the PIC to communicate with the Ground Station via a wireless serial transceiver. This communication link works at a baud rate of $115.2kbps$. The UART module is set up accordingly.

The UART module is also set up to request an interrupt whenever data is received from the Ground Station.

The second UART module is used for the serial communication between the GPS module and the PIC. This communication happens at a rate of $57.6kbps$. The UART module is set up to request an interrupt whenever data is received from the GPS module. This interrupt processes the GPS data as described in Paragraph 4.1.8.

4.1.8 GPS module

The GPS module has three very specific communication protocols. These protocols are described in reference [6]. The protocol standard used in this project is the UBX protocol standard. Figure 4.4 shows the structure of a UBX data packet.

These data packets are generated by the GPS receiver and sent to the PIC via the UART module, as described in Paragraph 4.1.7. The UART module requests an interrupt for every 8-bit data byte received through the UART module. A GPS parser function was written to analyse these received bytes and reconstruct UBX data packets. The data contained in the payload of the UBX data packet is allocated to the relevant variable. For instance, if the UBX data packet contains the latitudinal information then this information is stored in a variable named "Latitude".

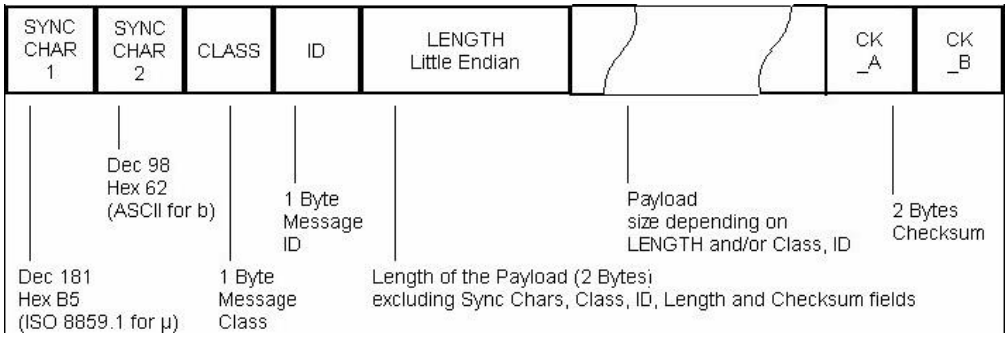


Figure 4.4: UBX Packet Structure (Source: [6])

4.2 Ground Station Software

The Ground Station Software was written with the Borland C++ Builder 6 Software. The Ground Station has a Graphical User Interface (GUI) to display all the incoming data concerning the sensors and actuators of the airship in real time. These data can be logged and stored as a binary file, and extracted at a later time for analysis.

The Control System is implemented in the Ground Station. The data received from the airship is used to calculate the control output to the actuators according to the sample period of the controller. A data package containing the actuator commands is sent back to the airship, in an effort to control the airship's heading and position.

It is important that the data transfer between the airship and the Ground Station is flawless. If any data is lost in the communication process, the airship could become uncontrollable, due to the corrupted data. The controllers will have a wrong feedback of what is happening on the airship and therefore the controller outputs will not be correct.

This paragraph explains all the technical parts of the Ground Station Software and how it is integrated to be an easy to use GUI, capable of monitoring and controlling the heading and position of an airship. This paragraph also explains how the Ground Station should be used and what every part on the GUI means.

Figure 4.5 shows the layout of the GUI.

4.2.1 Data Communication

There are different types of data the Ground Station can receive through two independent serial ports. They are the IMU data, the GPS data and the Camera data respectively. The IMU and the GPS units share the same serial port, while the Camera unit uses another serial port.

The IMU data consists of the measurements of the two-axis accelerometer, the gyroscope, the digital compass, and the two wind sensors. The GPS data consists of all the processed GPS data as received from the GPS satellites. This includes the time of the week, the latitude, the longitude and the height above sea level. The Camera data consists of a time unit, the image plane position and velocity of the airship centroid.

The Ground Station Software makes use of a State Machine for each serial port. Each 8-bit data byte received is compared to the data byte that the State Machine expected. The first two bytes are the signature of the data packet. Depending on the signature the State Machine knows how

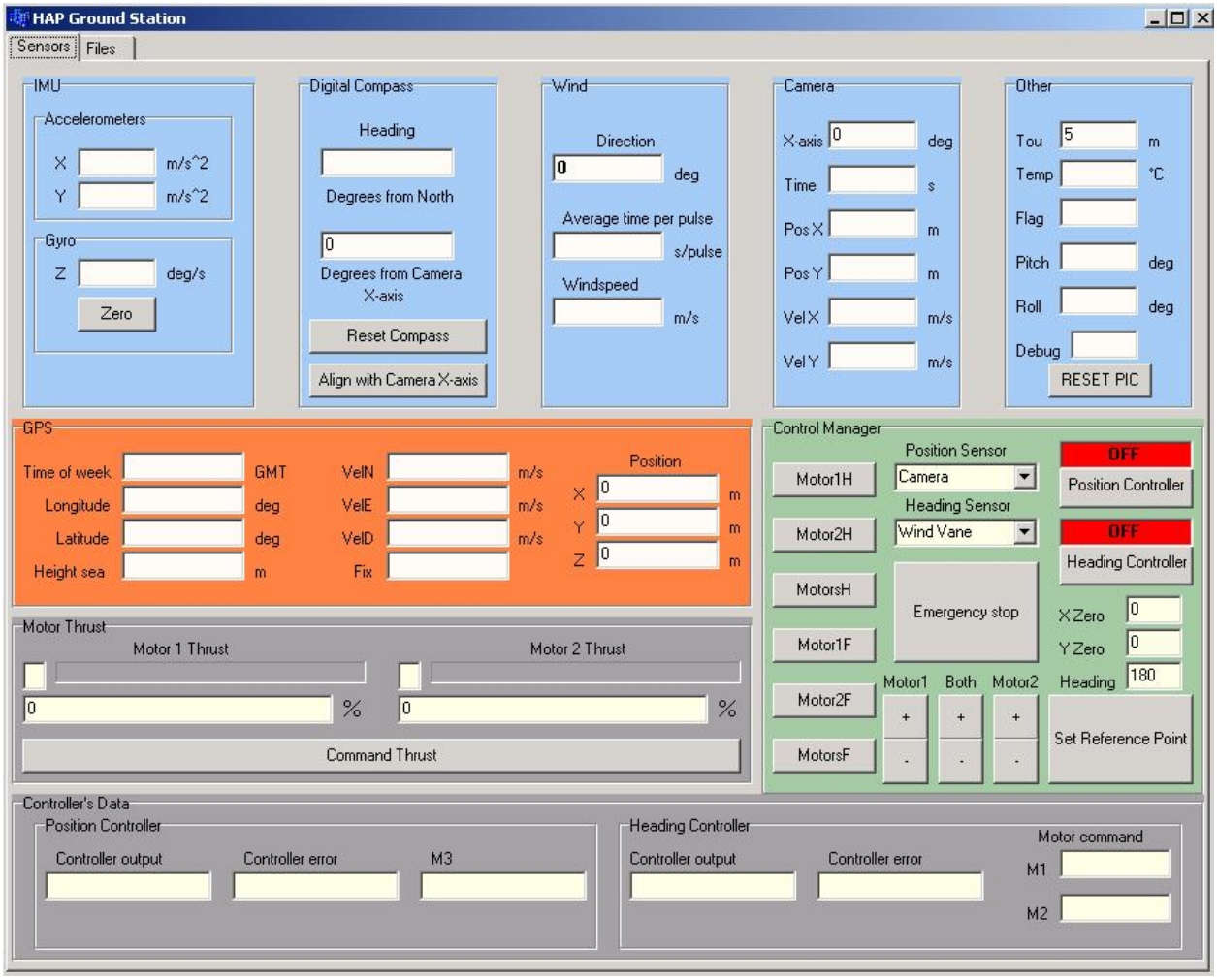


Figure 4.5: Ground Station: Sensors page

many data bytes are expected before the checksum bytes and the stop byte. The checksum bytes are calculated using the 8-bit Fletcher Algorithm, as described in [6]. If the checksum bytes are received correctly, it is assumed that the whole data packet was received correctly. Correct information can now be extracted from the data and displayed in the appropriate place on the Ground Station GUI. This data needs to be calibrated into appropriate units, as described in Paragraph 4.2.2.

The number of data packets that the State Machine receives and interprets correctly, is displayed in the GUI as well. In the unlikely event of an error occurring while receiving a data packet, the number of faulty data packets received is also displayed on the GUI. These faulty data packets are ignored by the controller.

4.2.2 Calibrating the Sensors

All the data packets received from the airship must be calibrated or scaled before being displayed in the Ground Station GUI.

For example: The accelerometer data is received as integer values between 0 and 1023, according to the values generated by the ADC. A value between 0 and 511 means a negative acceleration, 512 means the acceleration is zero, and a value between 513 and 1023 is a positive acceleration. When

the accelerometer is rotated so that one of the measuring axis is perpendicular to the surface of the earth, the measured acceleration must be $9.81 m/s^2$ according to the acceleration of the Earth's gravity. If the value received by the Ground Station in this situation is 567, the accelerometer is calibrated to give the correct acceleration of the airship in m/s^2, as:

$$Acceleration(m/s^2) = (0.155 \times Acceleration\_received) - 80.6 \tag{4.4}$$

The calibration of all the sensors is done in three respective functions in the Ground Station Software. These functions process the incoming data and store the calibrated data in corresponding variables.

4.2.3 Resetting the PIC

In order to synchronise the airship and the Ground Station, the PIC on board the airship must be reset. This is done manually by pressing the reset button on the circuit board of the airship, or by pressing the "RESET PIC" button on the Ground Station GUI. When the reset button is pressed, a command is sent to the airship via the serial port. The PIC receives this command and restarts the execution of the Software on the PIC, as described in Paragraph 4.1.

After resetting the PIC, the data packages will be received correctly from the PIC. All the data will be displayed in the relevant text boxes of the Ground Station GUI.

4.2.4 Resetting the digital compass

The digital compass must be reset to make sure the heading is measured correctly. This is done by pressing the "Reset Compass" button on the Ground Station GUI. This sends a command to the PIC which in turn sends a command to the digital compass according to the I^2C communication protocol, to reset the digital compass.

4.2.5 Aligning the axis systems

The next important thing that must be done before the Control Systems can be armed, is the alignment of the body axis of the airship with the inertial axis system of the camera. This is needed so that the positional coordinates received from the Camera Tracker can be transformed to the body axis of the airship. This is done by keeping track of the angle (Θ) of the x-axis of the airship relative to the x-axis of the Camera Tracker, as shown in Figure 4.6. This angle (Θ) is then used in a conversion matrix to transform from inertial coordinates to body coordinates. The conversion matrix is given in Equation 4.5.

$$\begin{bmatrix} x_{body} \\ y_{body} \end{bmatrix} = \begin{bmatrix} x_{cam}\cos\Theta + y_{cam}\sin\Theta \\ y_{cam}\cos\Theta - x_{cam}\sin\Theta \end{bmatrix} \tag{4.5}$$

This angle (Θ) is determined by the use of the digital compass. The x-axis of both the airship and the Camera Tracker are aligned when the "Align with Camera x-axis" button is pressed. The relative heading of the Camera Tracker's x-axis is now known, and the angle between the heading of the airship and the x-axis of the Camera Tracker can easily be calculated at any given time during a flight test.

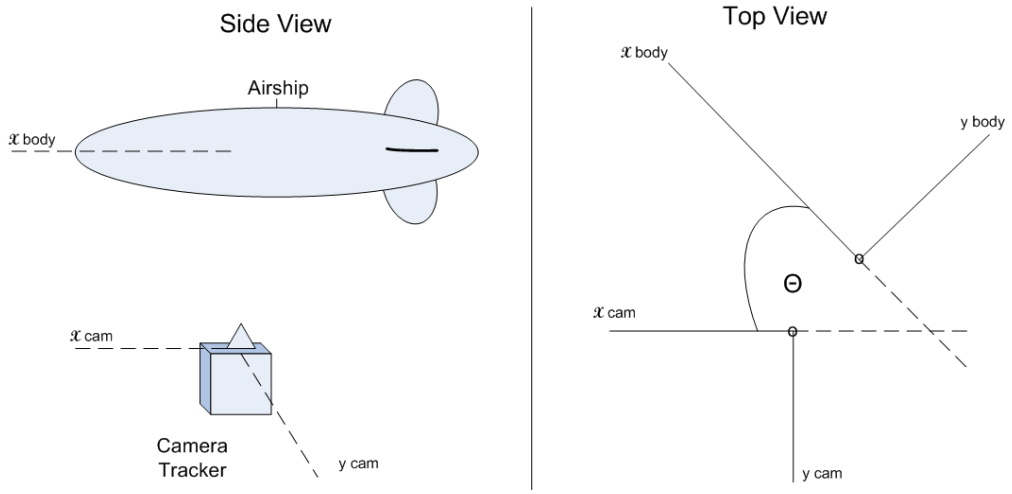


Figure 4.6: Relative angle between body axis and ground axis

If the Camera Tracker was aligned on the ground to magnetic North, which is the heading when the digital compass measures a 0° heading, it would mean that the camera and airship axis systems are already aligned.

4.2.6 Setting the Reference Point

The reference values of both the heading controller and the position controller are stored whenever the "Set Reference Point" button is pressed.

The position controller will try to control the position of the airship so that the airship will remain stationary at a specific position. This specified position is referred to as the reference point of the Position Controller. The default value of the reference point is at the centre of the Camera Tracker's field of view. The reference point however can be changed by pressing the "Set Reference Point" button on the Ground Station GUI. This will change the reference point to the position that the airship had when the button was pressed.

The reference position should be set to the position where the tether is vertical. This will ensure that the position controller only opposes the drag force created by the wind and do not need to oppose any lateral lift force component in the tether.

When the position controller was tested in Paragraph 5.2.5 the reference point was deliberately set to a position that was $1m$ away from where the tether was vertical. This was done to show how the position controller could oppose the lateral lift force component in the tether in order to maintain a certain position.

The reference value of the heading controller depends on which heading sensor was selected when the "Set Reference Point" button was pressed. If the wind vane was the selected heading sensor, the reference value will be the measured value of the wind vane. This will allow the heading controller to control the heading of the airship so that it follows the wind direction. If the digital compass was the selected heading sensor, the reference value could be any value between 0° and 360°. The heading controller will then aim to keep the heading of the airship at this value.

4.2.7 Activating the Controllers

Any one of the controllers can be activated separately, or both of them can be activated simultaneously. A controller is activated or deactivated by pressing the button of the relevant controller in the "Control Manager" block of the Ground Station GUI, as seen in Figure 4.7. Whenever a controller is activated, the colour of the text box above the controller changes from red to green. The opposite happens when a controller is deactivated.

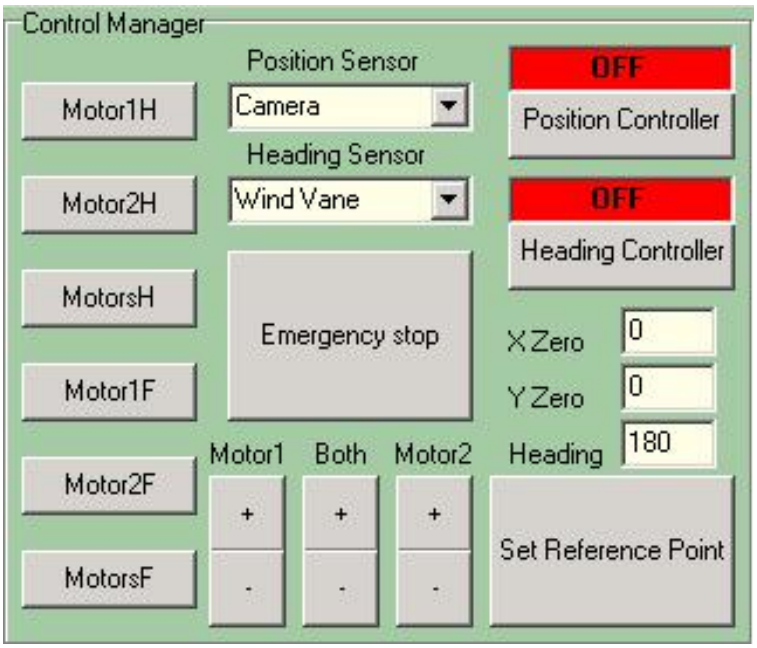


Figure 4.7: The "Control Manager" block of the ground station

The parameters of the Controllers will be displayed in the "Controller's Data" block while the controllers are activated, as seen in Figure 4.8. These parameters are updated according to the sampling time of the controller. A timer is used inside the Ground Station software and the output of the controllers are updated according to this timer.

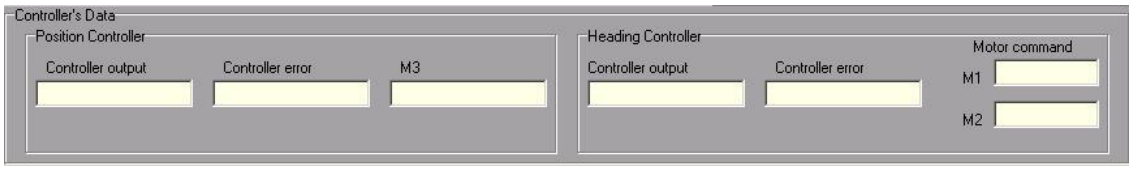


Figure 4.8: The "Controller's Data" block of the ground station

The outputs of the motors will be displayed in the "Motor Thrust" block in the Ground Station GUI, as seen in Figure 4.9.

4.2.8 Data Logging

All the Data can be logged and stored in a binary file. This is done in the "Files" page of the Ground Station GUI. Figure 4.10 shows how this page of the Ground Station GUI looks. The folder

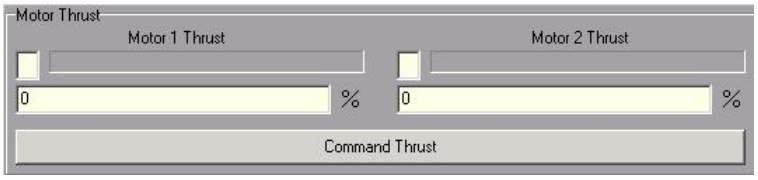


Figure 4.9: The "Motor Thrust" block of the ground station

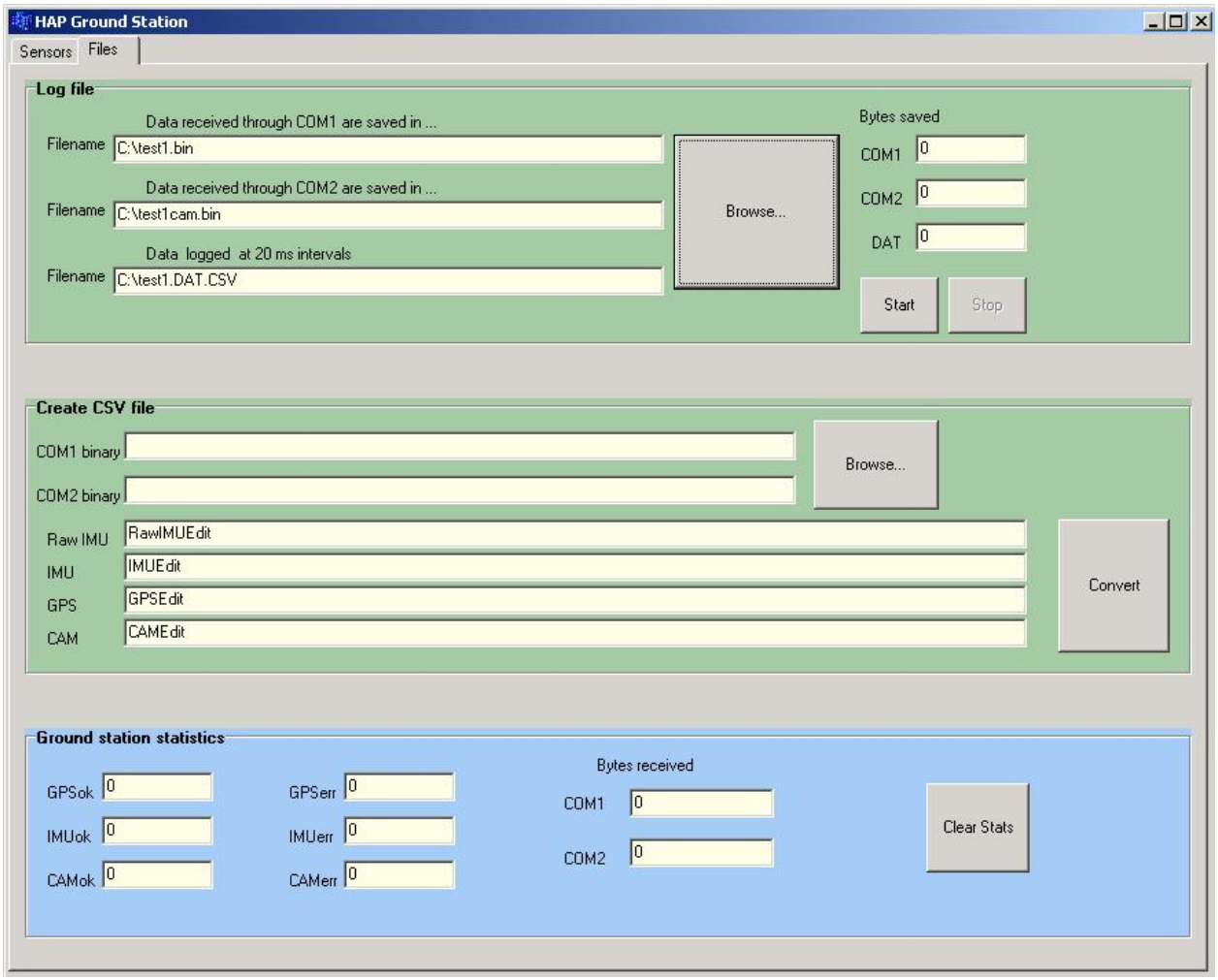


Figure 4.10: Ground Station: File handling page

and file name can be changed by pressing the "Browse" button in the "Log file" block. When the correct file name is selected, the "Start" button must be pressed to start data logging. The logging process will be terminated whenever the "Stop" button is pressed.

4.2.9 Creating CSV files

The logged data can be extracted at any later time for analysis, by creating a CSV file. To create a CSV file, the correct binary file must be loaded. Binary files can be selected by pressing the "Browse" button in the "Create CSV file" block of the Ground Station GUI and selecting the binary file that needs to be converted. The conversion will be done whenever the "Convert" button is pressed. This process will not work while new data is being received through any one of the serial

ports. The conversion creates four separate CSV files which contain the data for the different data streams separately. Paragraph 4.3 explains how these CSV files can be used to plot graphs of the various sensor and controller data.

4.3 Graph Plotting Software

The Graph Plotting Software was developed using MATLAB. A m-file was written to extract the data from the various data files generated by the Ground Station Software.

The name of the data file is inserted and the data is allocated to appropriate variables and organised into matrices from which graphs are plotted. The graphs visually show the behaviour of the airship during the specific flight test. The following graphs are drawn:

- A graph showing the position of the airship using the data received from the GPS module.
- A graph showing the position of the airship using the measurements of the Camera Tracker.
- A graph showing the velocity of the airship using the data received from the GPS module.
- A graph showing the velocity of the airship using the measurements of the Camera Tracker.
- A graph showing the acceleration of the airship using the measurements of the accelerometers.
- A graph showing the yaw rate of the airship using the measurements of the gyroscope.
- A graph showing the heading of the airship using the measurements of the digital compass.
- A graph showing the wind speed and wind direction disturbing the airship.
- A graph showing the pitch and roll angles of the airship as calculated from the accelerometer data.
- A graph showing the thrust forces applied by the motors to the airship.

The most important graphs are the heading, position and the applied thrust forces. These graphs are used mainly to analyse the performance of the airship controllers.

Chapter 5

Controller Design

This chapter explains how the hardware designed in Chapter 3 was used to control the airship system. The controllers were designed and tested on the dynamic model developed in Chapter 2. The controllers were then implemented on the airship and tested practically.

Two controllers were designed to control the airship. The first controller controls the heading of the airship. This controller is explained in Paragraph 5.1. The second controller controls the position of the airship. This controller is explained in Paragraph 5.2. The control signals of the two controllers are combined as in Paragraph 5.3 to control both the heading and the position of the airship simultaneously.

The results for the heading controller are shown in Paragraph 5.1.5, while the results of the position controller are shown in Paragraph 5.2.5.

5.1 Heading Controller

The heading controller applies a yaw-moment to the airship, to rotate the airship around its z-axis, for the airship to head into the wind. The tail fins of the airship will automatically weathercock the airship, but the heading controller will help to weathercock the airship faster and damp any yaw oscillations.

The digital compass is used to measure the current direction of the airship and the wind vane is used to measure the wind direction. The measurements of the wind vane are used to determine the reference flight angle. The difference between the actual heading of the airship and the current wind direction gives an error signal to the heading controller. This error signal should be made zero by applying the correct control signal to the actuators of the airship to rotate the airship until it settles at the correct heading.

The amplitude of the applied yaw-moment will depend on the control signal of the heading controller. The control signal is the output of the heading controller and is given in Newton meter (Nm) units. A yaw-moment is practically commanded to the airship, by applying a forward thrust to one of the brushless DC motors and an equivalent backward thrust to the other brushless DC motor. The control signal is scaled so that the value in Newton meters is transformed to a corresponding percentage of the total yaw-moment which the motors can give. The maximum yaw-moment which these motors can apply, was practically measured as $1.8Nm$. This means, for example, that a control signal of $0.9Nm$ would be scaled to 50% of the maximum available yaw-moment. The one

motor will therefore be commanded to give a backward thrust of 50% and the other motor will be commanded to give a forward thrust of 35%. A maximum yaw-moment is applied by commanding a backward thrust of 100% to the one motor and a forward thrust of 70% to the other motor. The reason why the two thrust magnitudes differ in size is because of the propeller efficiency, which differs between the forward thrust and the backward thrust. The propellers are only 70% effective when a backward thrust is applied as opposed to a forward thrust. The combination of the applied thrusts will give the correct yaw-moment to the airship.

The heading controller was implemented in the Ground Station computer, presented in Paragraph 5.1.4. Due to this, it was necessary to design a digital controller. The *Digital Control of Dynamic Systems* textbook [27] was used as an aid in designing the controller.

The first step in designing the digital heading controller was to identify a linear model describing the yawing motion of the airship. The system identification results are presented in Paragraph 5.1.1.

The digital heading controller was designed using a root locus design method. The root locus design is described in Paragraph 5.1.2.

The digital control system was first tested using the simulation of the nonlinear airship model. The digital control system was also implemented in the Ground Station computer and tested practically on the airship, as described in Paragraph 5.1.4.

5.1.1 System Identification of Heading Model

The system identification can either be done practically or by executing simulations of the non-linear airship model derived in Chapter 2. In this thesis the tests were done practically and the results were confirmed by the use of the simulation software.

The main idea behind the practical tests was to obtain the response of the airship to various yaw-moment pulses. The system identification tests were also done in a wind free environment. Yaw-moment pulses were applied to the airship and varied in amplitude and length. The response of the airship was obtained through analysing the data recorded by the sensors on the gondola. The data was analysed through relevant graphs drawn by the graph plotting software of Paragraph 4.3.

Figures 5.1 to 5.3 show the response of some of the yaw-moment pulses applied to the airship system. These figures show the amount of degrees through which the airship rotated in response to an applied yaw-moment pulse. The applied pulse is drawn on the same graph so that it is easier to see when the pulse was applied. The applied pulse is given as a percentage of the maximum yaw-moment. The angle through which the airship rotated was measured by the digital compass.

Table 5.1 gives a summary of the results.

Length of pulse	Applied yaw-moment	Amplitude of rotation
$6s$	1.8 Nm	34°
$11s$	1.8 Nm	50°
$20s$	1.8 Nm	100°

Table 5.1: Summary of practically measured yaw-moment pulse responses

The results were used to obtain the following linear model to describe the yawing motion of the airship:

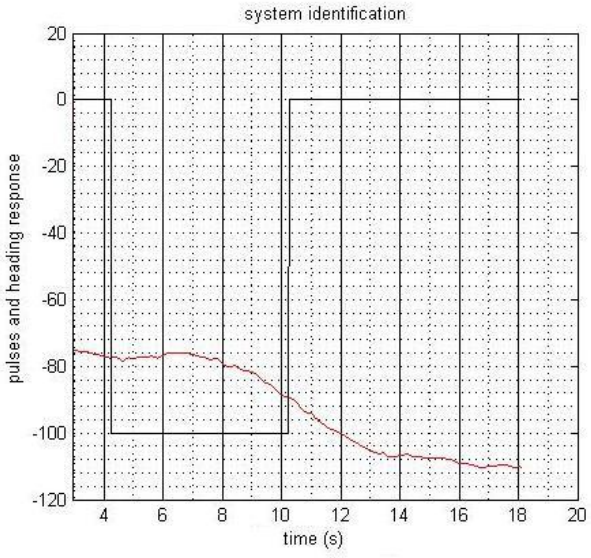


Figure 5.1: Response to a 1.8 Nm yaw-moment, applied for 6 seconds

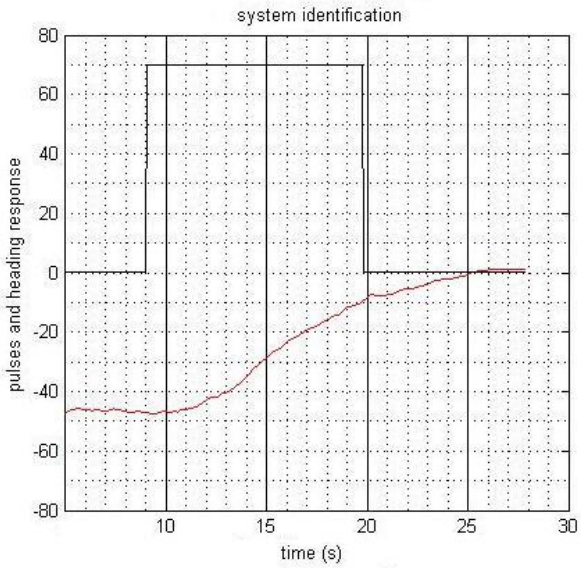


Figure 5.2: Response to a 1.8 Nm yaw-moment, applied for 11 seconds

$$G(s) = \frac{Y(s)}{U(s)} = \frac{2.5}{s^2 + 0.7s} \tag{5.1}$$

Where, $Y(s)$ is the yaw angle Laplace function (units in degrees) and $U(s)$ is the applied yaw moment Laplace function (units in Nm).

Figures 5.4 to 5.6 show the response of the linear model in Equation 5.1 when similar pulses are applied to the simulation model as to the airship. The applied pulse is drawn on the same graph together with the heading response. In these figures the applied pulse is given in Newton meter and not as a percentage of the maximum yaw-moment.

The linear model in Equation 5.1 was transformed to its discrete equivalent using a sampling time (T_s) of $0.5s$. This was done using MATLAB. Equation 5.2 shows the discrete transfer function

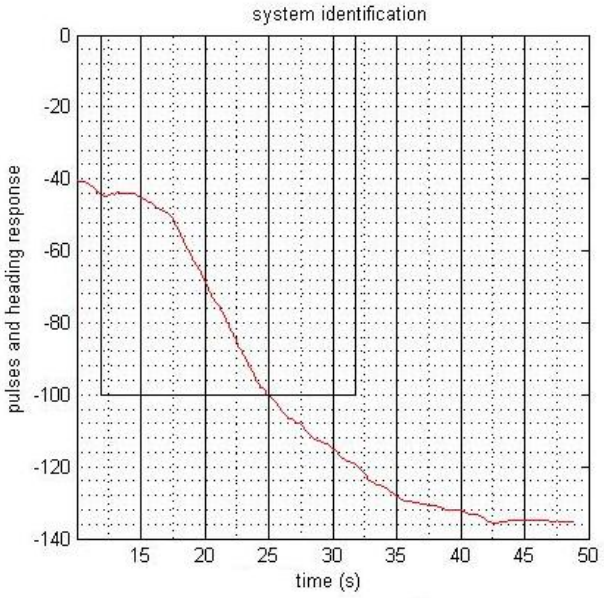


Figure 5.3: Response to a 1.8 Nm yaw-moment, applied for 20 seconds

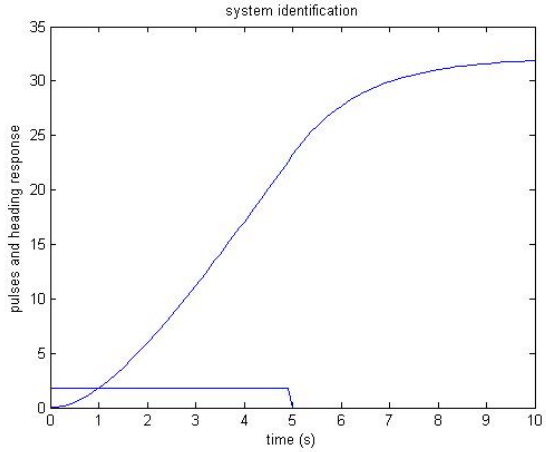


Figure 5.4: Simulation results of a 1.8 Nm yaw-moment, applied for 5 seconds

of the continuous model with zero order hold.

$$G(z) = \frac{0.279z + 0.2483}{z^2 - 1.705z + 0.7047} \qquad (T_s = 0.5s) \tag{5.2}$$

A closer look at Figures 5.1 to 5.3, shows that there exists a delay of approximately two seconds between the time the yaw-moment is applied to the airship and the time the airship actually starts to rotate. This delay can be modelled into the discrete linear model by adding additional poles at the origin of Equation 5.2. Every pole added to the discrete model creates a delay equal to the sampling period used. This means that four sample delays of $0.5s$ each must be added to the discrete linear model to obtain a delay of two seconds.

The reason why the delay exists in the system is due to the time it takes the motors to reach the required rotation rate and the maximum thrust for that rotation rate. The delay differs for different applied yaw-moments. It was however decided that a constant delay of two seconds should

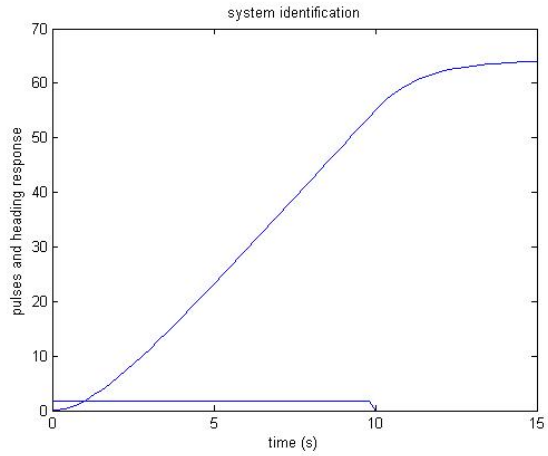


Figure 5.5: Simulation results of a 1.8 Nm yaw-moment, applied for 10 seconds

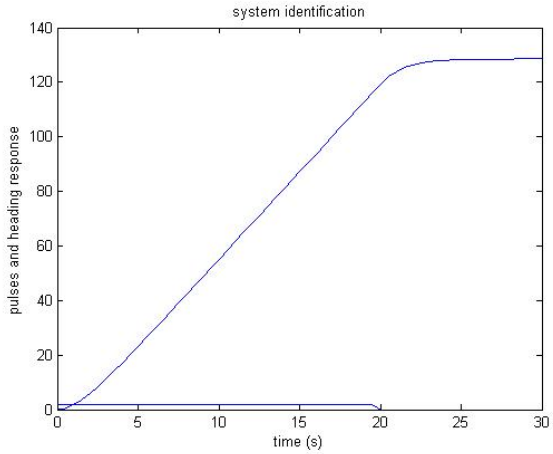


Figure 5.6: Simulation results of a 1.8 Nm yaw-moment, applied for 20 seconds

be sufficient to design an accurate model.

Equation 5.2 becomes a sixth order transfer function due to the addition of these delays. This sixth order transfer function is given in Equation 5.3.

$$G(z) = \frac{0.279z + 0.2483}{z^6 - 1.705z^5 + 0.7047z^4} \qquad (T_s = 0.5s) \tag{5.3}$$

Paragraph 5.1.2 shows how a controller can be designed and implemented for the heading model.

It is more difficult to design an effective controller with dominant closed loop poles for higher order discrete linear models. An alternative to adding four poles at the origin of the transfer function in Equation 5.2 can be considered in an effort to decrease the order of the discrete linear model: The linear model in Equation 5.1 can be transformed to a discrete equivalent with a sampling period of $1.0s$ instead of the sampling period of $0.5s$ as used before. Only two extra poles will be needed in the discrete linear model to model the two second delay, which will restrain the discrete linear model to a fourth order open loop heading transfer function. This means that the linear model in Equation 5.1 must be recalculated in MATLAB. This time a sampling period of $1.0s$ is used, which leads to the discrete linear model in Equation 5.4.

$$G(z) = \frac{z + 0.795}{z^2 - 1.497z + 0.497} \qquad (T_s = 1.0s) \tag{5.4}$$

Equation 5.4 changes to Equation 5.5 when the two poles are added to incorporate the two second delay.

$$G(z) = \frac{z + 0.795}{z^4 - 1.497z^3 + 0.497z^2} \qquad (T_s = 1.0s) \tag{5.5}$$

Paragraph 5.1.2 shows how a controller can be designed for this fourth order discrete linear model.

The practical results for the designed controllers are presented in Paragraph 5.1.5.

5.1.2 Root Locus Design of Heading Controller

A root locus is a graphical illustration for the closed loop poles (roots of the characteristic polynomial) that change as the loop gain is varied. By applying some rules, the root locus plot can be used to design a closed-loop system with specific characteristics. [27]

This thesis looks at the design, implementation and practical results of two different heading controllers. The sampling periods of the two heading controllers differ and use the two open loop heading models of Equations 5.3 and 5.5 respectively. The complexity of the two heading controllers also varies: The first controller uses only one lag-network in the compensator and doesn't have an integrator. The second controller consists of an integrator which is part of a lag-network and two lead-networks.

Other heading controller designs could also exist. The aim of this thesis was not to design the optimal heading controller, but to design an effective controller with good practical results.

Discrete root locus design for model with $T_s = 0.5s$

Figure 5.7 shows the root locus design for the sixth order discrete linear model of Equation 5.3. A lag-network controller was designed for this model. Lag networks can approach integral control and tend to improve the steady-state error of the system, according to [27].

The open loop heading model of Equation 5.1 was used to design this controller. The controller is designed to have the following closed loop specifications: A closed loop damping coefficient (ζ) of 0.7 and a 1% settling time (t_s) of the closed loop step response of $25s$. The closed loop poles satisfying these closed loop specifications are calculated as:

$$\omega_n = \frac{4.6}{\zeta t_s} = 0.263 rad/s \tag{5.6}$$

$$z_{CL} = e^{-\zeta\omega_n T_s} \angle(\pm\omega_n T_s \sqrt{1 - \zeta^2}) \tag{5.7}$$

Solving Equation 5.7 gives dominant closed loop poles at $z_{CL} = 0.912\angle(\pm 0.094)$. This can also be written as: $z_{CL} = 0.908 \pm j0.086$.

The zero and the pole of the lag-network can be chosen to ensure that the root locus passes through the desired location of the closed-loop pole position. The gain can now be tweaked to

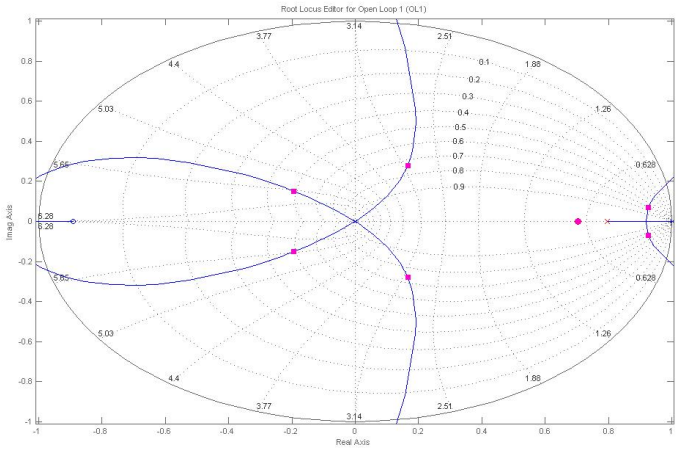


Figure 5.7: Root locus design ($T_s = 0.5s$)

obtain the correct dominant closed-loop poles. The discrete transfer function of the lag-network controller, as designed for this discrete linear model, is given as:

$$D(z) = 0.025\frac{z - 0.704}{z - 0.796} \qquad (T_s = 0.5s) \tag{5.8}$$

The step-response of this closed-loop discrete linear system, as well as the control signal needed to obtain this response is shown in Figure 5.8.

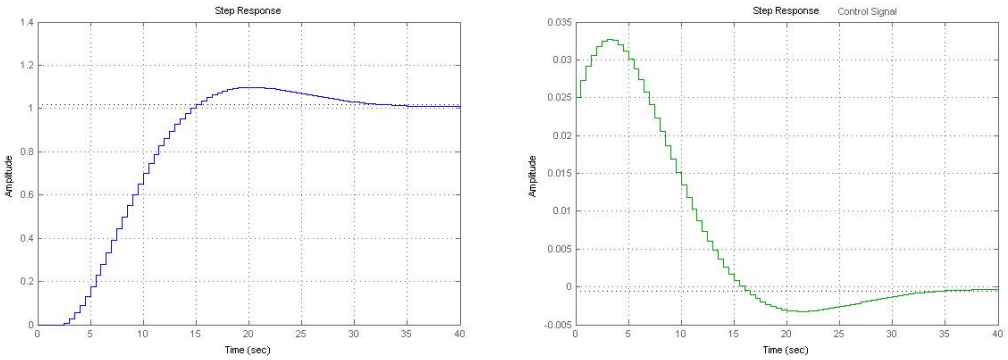


Figure 5.8: Step response and control signal of closed-loop system ($T_s = 0.5s$)

The results obtained practically when testing this controller, are shown and discussed in Paragraph 5.1.5.

Discrete root locus design for model with $T_s = 1.0s$

Figure 5.9 shows the root locus design for the fourth order discrete linear model of Equation 5.5. The controller designed for this model consists of two lead networks and a lag network with an integrator.

The procedure for designing this controller can be described as follows: Firstly an integrator is added to the root locus design. An integrator in the controller will ensure zero steady state errors

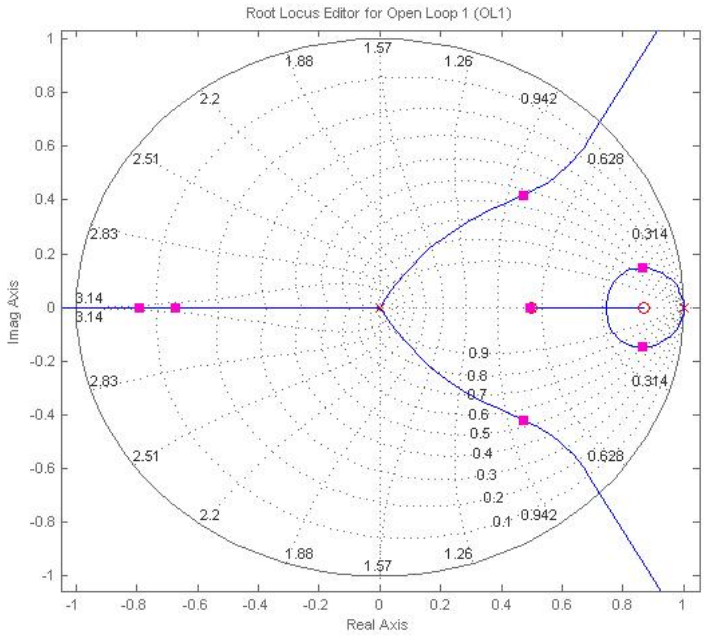


Figure 5.9: Root locus design ($T_s = 1.0s$)

to the constant reference inputs to the system. The next step is to cancel the open-loop pole of the model with a controller zero. This completes the lag-network. The two lead-networks can be designed by placing two poles at the origin of the root locus. The zeros that accompany the poles of these lead-networks and the controller gain can now be designed until the step-response of the closed-loop system is acceptable. The controller designed for the heading system, is:

$$D(z) = 0.27679\frac{(z-0.606)(z-0.496)(z-0.868)}{z^2(z-1)} \qquad (T_s = 1.0s) \tag{5.9}$$

The step-response of this closed-loop discrete linear system, as well as the control signal are shown in Figure 5.10.

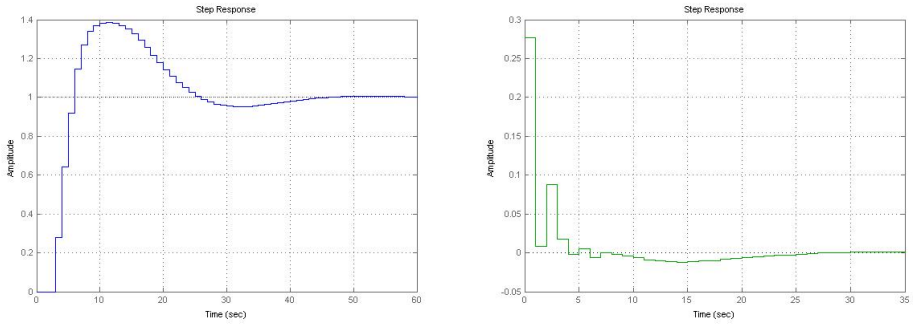


Figure 5.10: Step response and control signal of closed-loop system ($T_s = 1.0s$)

It is interesting to note how the control signal in Figure 5.8 differs from the control signal in Figure 5.10. The step responses of the heading controllers are almost the same, although the $1.0s$

controller has slightly more overshoot than the $0.5s$ controller but the control signals of the two controllers differs significantly. It seems that the amount of control effort that is needed to control the heading of the airship differs from one controller to the next.

The results obtained practically when testing this controller are shown and discussed in Paragraph 5.1.5.

5.1.3 Simulation Results of Heading Control

Before the control systems were implemented and tested on the airship system, they were verified in simulation. A heading controller was added to the nonlinear model of the airship created in Chapter 2.

The heading output of the airship model was sampled according to the sampling period of the controller. This was done by adding a zero order hold circuit to the heading output of the airship model.

The heading error was calculated by subtracting the heading output from the reference angle input. The calculated heading error is the input to the heading controller.

The control signal output of the heading controller was connected to the thrust inputs of the airship model. The control signal was limited according to the maximum and minimum control effort of the actuators.

It was seen in practise that the motors have a dead-band region between -5% and 5% of the total thrust. This property of the actuators was added to the simulation in order to make the simulation more accurate.

Figure 5.11 shows the Simulink blockdiagram of the simulation with the added heading controller.

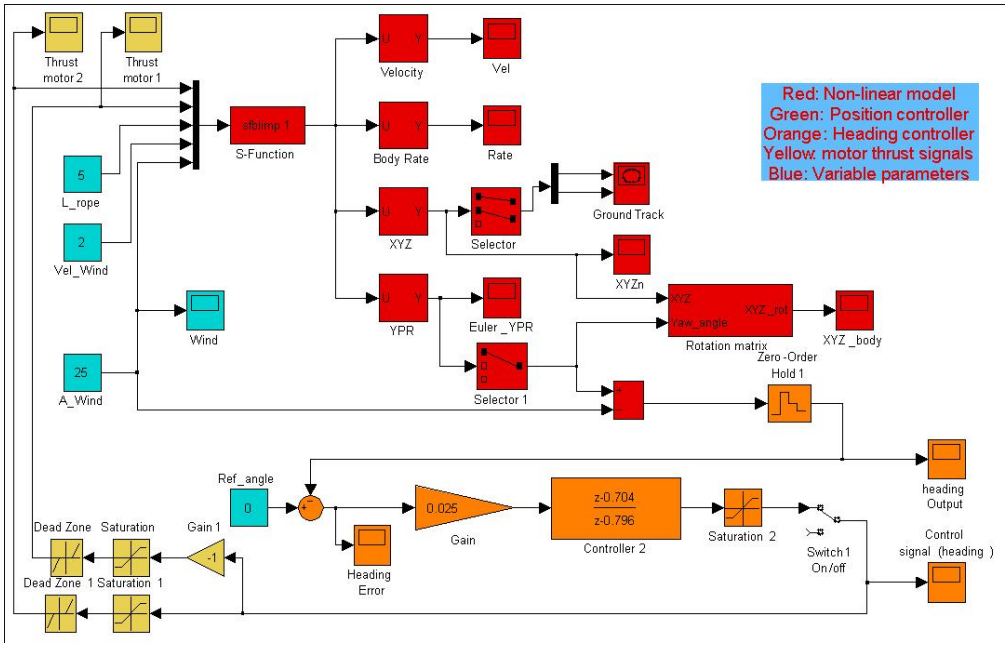
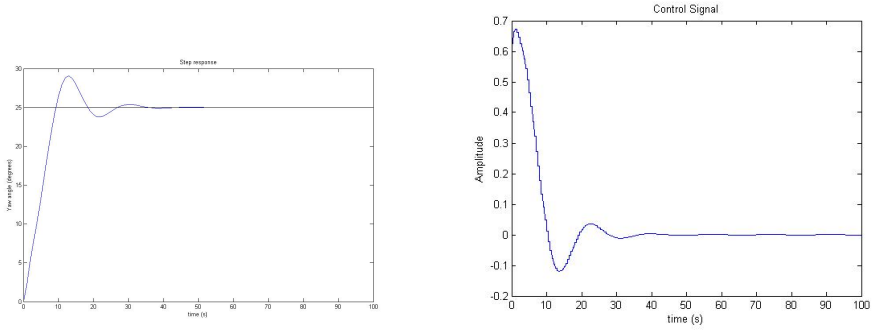


Figure 5.11: Blockdiagram of heading controller in simulation

Figure 5.12 shows the results of a simulation executed to illustrate the working of the heading controller. Figure 5.12 uses the heading control system designed with a sampling period of $T_s = 0.5s$.



(a) Step response of heading (b) Control signal of heading controller

Figure 5.12: simulation results of $0.5s$ heading controller

The simulation results in Figure 5.12 can be compared to the closed loop step response of the root locus design shown in Figure 5.8. The simulation used the non-linear airship model, while the step response of the root locus design used the linearized heading model of the airship. The step response and control signal of both figures are very similar. This is an indication that the linearized heading model is a good approximation of the non-linear model. The results verify that this heading controller should work in practise and can now be implemented and tested.

5.1.4 Implementation of Heading Controllers

In order to implement the designed heading controllers on a computer or a micro-processor, it was necessary to rewrite the discrete transfer functions of the heading controllers as difference equations. Equation 5.10 shows that the numerator of a transfer function corresponds to the output terms of the transfer function, while the denominator corresponds to the input terms of the transfer function. The transfer function in Equation 5.8 can therefore be written as output terms and input terms respectively.

$$\begin{aligned} D(z) = 0.025\frac{z-0.704}{z-0.796} \times \frac{z^{-1}}{z^{-1}} &= \frac{U(z)}{E(z)} \\ U(z) \times (z-0.796)z^{-1} &= E(z) \times 0.025(z-0.704)z^{-1} \\ U(z) \times (1-0.796z^{-1}) &= E(z) \times 0.025(1-0.704z^{-1}) \end{aligned} \tag{5.10}$$

The inverse Z-transform of Equation 5.10 can now be taken to give the following difference equation:

$$u_1(k) = 0.796u(k-1) + 0.025(e(k) - 0.704e(k-1)) \qquad (T_s = 0.5s) \tag{5.11}$$

The same can be done with the transfer function in Equation 5.9. This is shown in Equation 5.12.

$$D(z) = 0.27679\frac{(z-0.606)(z-0.496)(z-0.868)}{z^2(z-1)} \times \frac{z^{-3}}{z^{-3}} = \frac{U(z)}{E(z)}$$
$$U(z) \times (z^3 - z^2)z^{-3} = E(z) \times 0.27679(z^3 - 1.97z^2 + 1.25z - 0.261)z^{-3}$$
$$U(z) \times (1 - z^{-1}) = E(z) \times 0.27679(1 - 1.97z^{-1} + 1.25z^{-2} - 0.261z^{-3}) \quad (5.12)$$

The inverse Z-transform of Equation 5.12 can now be taken to give the following difference equation:

$$u_1(k) = u(k-1) + 0.27679\big(e(k) - 1.97e(k-1) + 1.25e(k-2) - 0.261e(k-3)\big) \quad (T_s = 1.0s) \quad (5.13)$$

The term $u_1(k)$ correlates with the new control signal, which is the output of the heading controller. This output value is calculated every sample period. The $u(k-1)$ and $u(k-2)$ terms correlate with the previous two control signal output values. The $e(k)$ term correlates with the newest value of the measured error and the $e(k-1)$, $e(k-2)$ and $e(k-3)$ terms correlate with the previous three values of the measured error. The new value of the control signal ($u_1(k)$) is therefore dependant on the previous values of the control signal as well as the current and previous values of the error value. The error value is measured once every sampling period. The old values of the error and control signal are shifted into the corresponding registers and the new control signal is calculated according to the difference equation. This procedure is executed every sampling period.

The control signal is applied to the actuators in an effort to command the correct yaw-moment to the airship. To create a yaw-moment, the sign of the control signal that is applied to each actuator should be opposite to each other. A positive yaw-moment will require the thrust of $motor_1$ to be positive and the thrust of $motor_2$ to be negative. The inverse is true for a negative yaw-moment.

The heading controller uses the measurement of the digital compass to measure the heading of the airship. The measured error is the difference between the reference angle and the measurement of the digital compass. The reference angle can either be a constant value or it can be the angle measured by the wind vane.

The control signal should not exceed the maximum control effort which the actuators can deliver to the system. Therefore the control signals should be limited to the maximum and minimum values which can be applied to the actuators. If the control signal is limited to these values, it will also prevent integrator windup from occurring.

The heading controller was practically implemented in C++ in the Ground Station software. This software is explained in Paragraph 4.2.

5.1.5 Practical Results of Heading Controller

The two heading controllers designed in Paragraph 5.1.2 have been implemented and tested separately, under identical conditions. The heading controllers were tested indoors where no wind was present. The altitude of the airship was limited with a $5m$ long tether. Extra weight was added to the airship so that the excess lift of the airship was less than $1kg$. The results of both controllers are given in this chapter.

The heading controllers were tested by measuring different step responses. The accuracy by which the airship followed the given reference angle was an indication of the effectiveness of the specific controller. The time it took the heading of the airship to settle at the reference angle was another indication of the effectiveness of the specific controller.

Results for model with $T_s = 0.5s$

Figure 5.13 shows the step response of the heading of the airship for a constant reference angle of $0°$. The initial heading of the airship, as measured by the digital compass, was $-50°$. It was seen that it took the controller approximately $20s$ to reach the reference angle. The heading output showed that the airship oscillates slightly around the reference angle. The amplitude of this oscillation was approximately $6°$ and the frequency thereof was approximately $0.025Hz$.

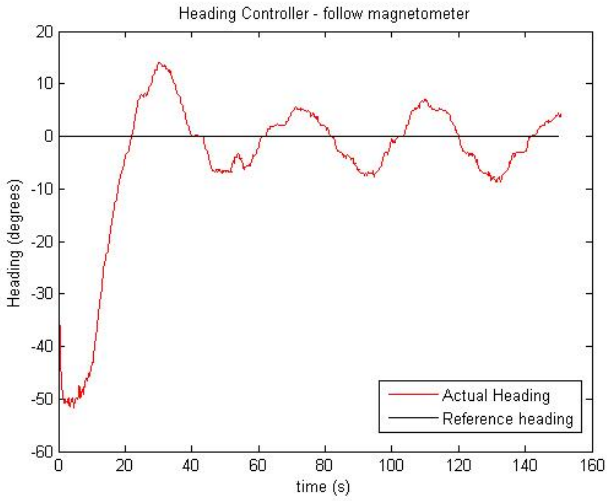


Figure 5.13: Step response of heading controller ($T_s = 0.5s$)

Figure 5.14 shows the control signal applied to the actuators to give the response of Figure 5.13. The control signal for each brushless DC motor is given. It is seen that the control signal applies opposite thrusts to each motor. This created the necessary yaw-moment to rotate the airship. Note that the negative thrust was always bigger than the positive thrust. This is due to the efficiency of the propellers that vary between forward and backward thrusts.

Figure 5.15 shows the step response of another test done with the same heading controller. This time the heading reference was changed to different values during the test. The oscillations of the heading output is still present, but this figure shows that the oscillations are in fact quite small and that the heading controller follows the reference heading quite accurately.

Figure 5.16 shows the respective thrusts applied by the actuators for this test.

The last test performed with this heading controller was a step response that used the measurements of the wind vane as the reference angle of the heading controller. Figure 5.17 shows the results for this test.

The results from this test show the oscillations of the heading output of the airship around the reference heading. It is seen that the measured reference angle of the wind vane is quite noisy. This is due to the long cable that is used to connect the wind vane to the airship. The measurements

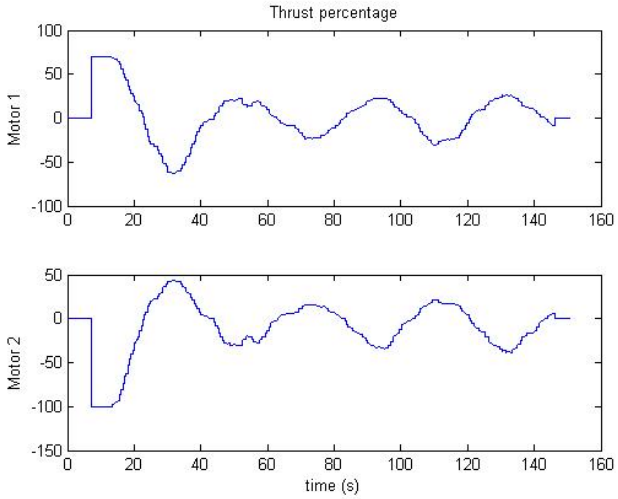


Figure 5.14: Control signals of both actuators ($T_s = 0.5s$)

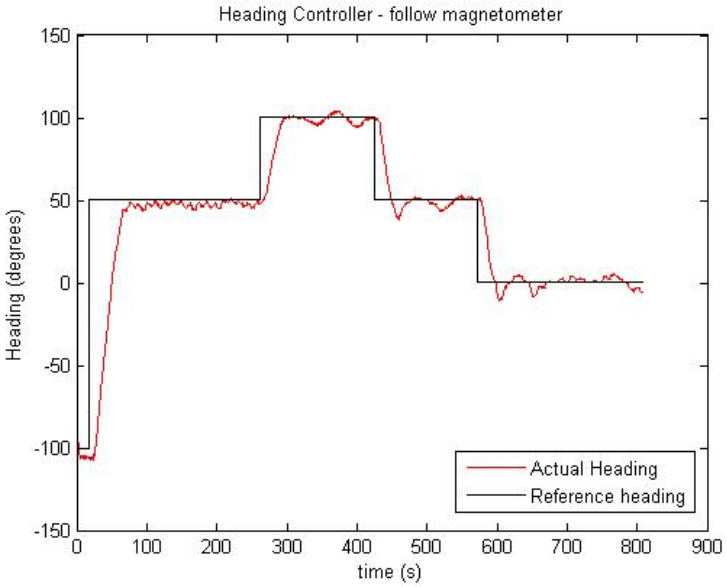


Figure 5.15: Step response of heading controller ($T_s = 0.5s$)

of the wind vane are filtered in software, but it could be improved further by using better quality cable to connect the wind vane to the airship.

Results for model with $T_s = 1.0s$

Figure 5.18 shows the step response of the heading of the airship for a reference angle given by the digital compass as 0°. The initial heading of the airship was at $-120°$. It is seen that it takes this heading controller approximately $40s$ to reach the reference angle. When comparing Figure 5.18 with Figure 5.13, it can be seen that the heading output of this heading controller doesn't oscillate around the reference angle, as was seen with the previous heading controller. This heading controller follows the reference angle better than the previous controller. The time it takes the controller to settle at the reference angle is approximately the same for both heading controllers.

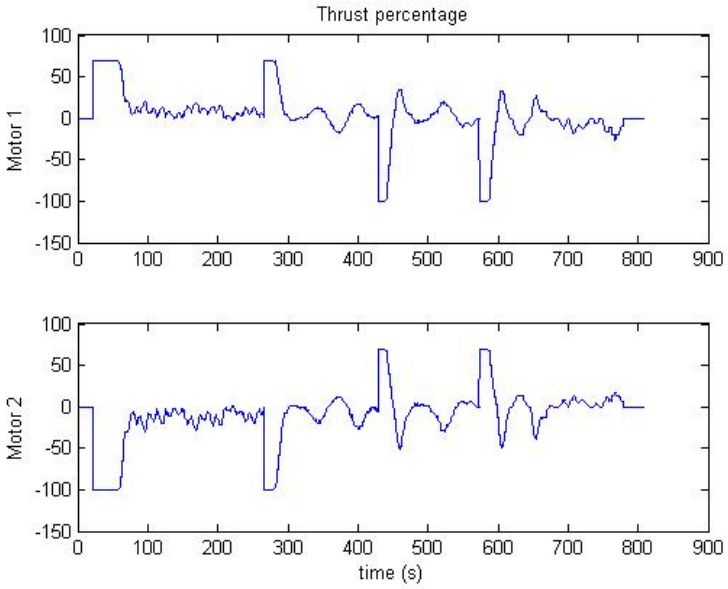


Figure 5.16: Control signals of both actuators ($T_s = 0.5s$)

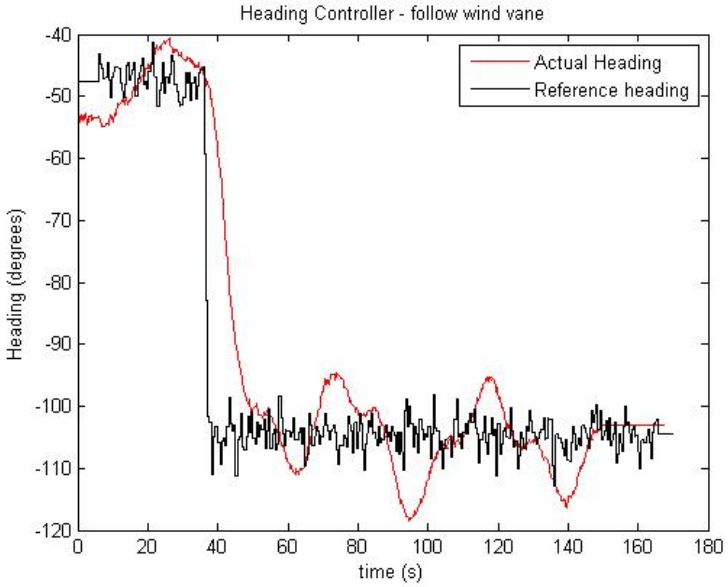


Figure 5.17: Step response of heading controller ($T_s = 0.5s$)

The same heading controller was also tested in nominal wind conditions. The airship was again commanded to follow a reference angle as given by the digital compass. Any wind disturbances should be corrected and a reference heading of 0° should be maintained. The results of this practical test are given in Figures 5.19 and 5.20.

The decision of which heading controller to use in the end could be influenced by various factors: The accuracy by which the controller follows a reference angle must be considered together with the amount of power that the controller will consume in an effort to follow that reference angle. The second controller is more accurate than the first controller. The power consumption of the controllers depends on the amount of control effort it uses to control the heading of the airship. The

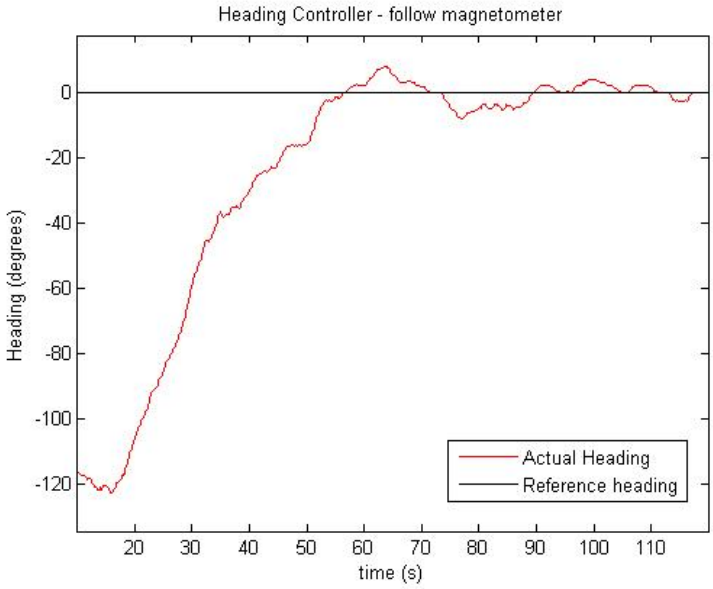


Figure 5.18: Step response of heading controller ($T_s = 1.0s$)

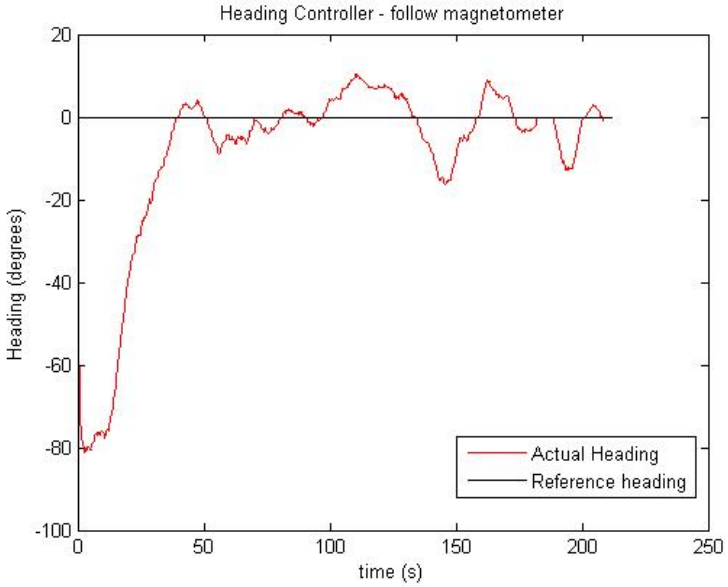


Figure 5.19: Step response of $1.0s$ heading controller in nominal wind conditions

second controller uses more control effort in order to be more accurate than the first controller. The complexity of the controller could also have an effect on the decision, because complex controllers are more difficult to design. The first controller is less complex than the second controller. A perfect controller might not exist, but some effort can be made to find an optimal controller that gives a good balance between accuracy, power consumption and simplicity.

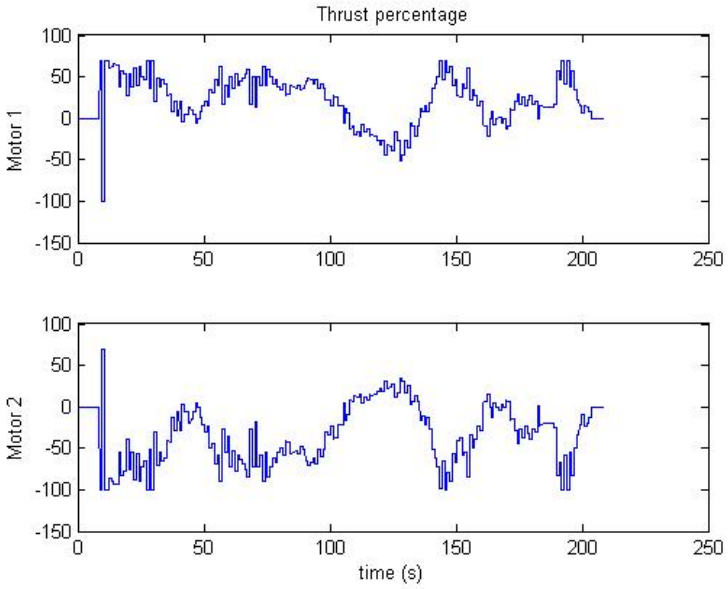


Figure 5.20: Control signals of both actuators in nominal wind conditions

5.2 Position Controller

The next step was to design a position controller. The function of the position controller is to oppose the drag force created by the wind. The wind will push the airship away from its original position. The heading controller will align the nose of the airship with the wind direction while the position controller will manoeuvre the airship against the wind in order to maintain a certain position. The two actuators of the airship will be used to generate a forward thrust, along the body x-axis, that will oppose the drag force created by the wind. A forward thrust is commanded to the airship by applying a positive thrust to each actuator. The magnitude of the thrust in each actuator should be equal.

The tether will influence the movement of the airship: When the wind moves the airship so that the tether is not vertical anymore, then a force in the tether will pull the airship in the opposite direction of the drag force. This will make the airship move back and forth in the direction of the wind. The position controller should damp this oscillation so that the tether will eventually become vertical again and the horizontal force of the tether will be zero.

The position of the airship is determined by the Camera Tracker, presented in Paragraph 3.3. The position of the airship where the tether is vertical will be the reference position. The position error is determined by calculating the difference between the actual position and the reference position.

The first step in designing the position controller was to identify the open loop characteristics of the airship. This is explained in Paragraph 5.2.1. This resulted in a linear position model that approximated the movement of the airship for different wind conditions.

The next step was to design a discrete controller according to some closed loop characteristics. This was done by the use of a root locus design method, presented in Paragraph 5.2.2.

This controller was tested in simulation and verified in practise. The practical test results of the position controller are given in Paragraph 5.1.5.

5.2.1 System Identification of Position Model

The system identification tests can either be done practically or it can be simulated by use of the non-linear airship model derived in Chapter 2. In this case the tests were done practically with a $5m$ long tether and the results confirmed by the use of a simulation. The simulation was used to expand the results for system identification tests that could not be done practically. This was done for all tether lengths unequal to $5m$.

The system identification tests were done in controlled wind conditions. The airship was headed into a controlled wind, generated by a big fan. The velocity of the wind was varied according to three different speed settings of the fan. The movements of the airship in the various winds were recorded so that a linear position model could be approximated from the results.

A typical test starts with the airship at its position of origin with a vertical tether. A constant wind is applied as a step-input that will push the airship in the opposite direction of where the wind is coming from. The airship will continue to move in this direction until the force in the tether starts to pull the airship back against the direction of the wind. The airship will oscillate according to the drag force and the force in the tether. The oscillations will be damped slowly and the airship will settle at a new position where the drag force and the opposing force in the tether is equal. The position at which the airship settles is called the "Offset" in Table 5.2 and the magnitude thereof depends on the velocity of the wind.

Figure 5.21 shows the simulated movement of the airship in a $1m/s$ wind and a $5m$ tether.

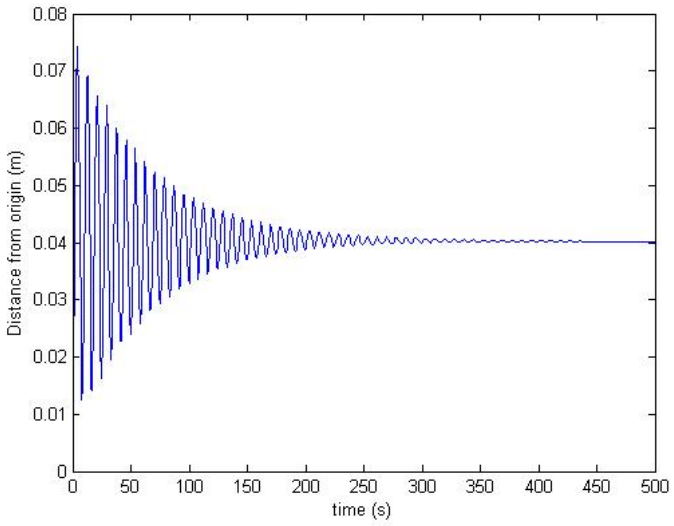


Figure 5.21: Response to a constant wind of 1 m/s ($tether = 5m$)

Table 5.2 gives a summary of the results obtained from multiple practical tests and simulation results. The velocity of the wind in Table 5.2 was constant at $1m/s$.

These results led to an approximation of a linear position model that describes the movement of the airship against the wind. The linear position model can be described as a second order linear transfer function, as:

Length of tether	**Offset (K)**	**Natural frequency (ω_n)**	**Damping coefficient (ζ)**
$5m$	$0.04m$	$0.48rad/s$	0.03
$10m$	$0.08m$	$0.35rad/s$	0.03
$15m$	$0.12m$	$0.28rad/s$	0.03
$20m$	$0.16m$	$0.25rad/s$	0.03

Table 5.2: Results of system identification tests for position controller (wind velocity = 1 m/s)

$$G(s) = \frac{K\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{5.14}$$

The parameters in Equation 5.14 correspond with the parameters given in Table 5.2. The table shows that the linear model of the airship changes according to the length of the tether. It is necessary to design different controllers for different tether lengths. The controllers were designed for a wind velocity of $1m/s$.

The offset and the damping coefficient parameters of the linear model increase when the velocity of the wind increases. This can be seen by comparing Table 5.2 with Table 5.3. The controllers were designed for the low damping coefficients and low offsets as given in Table 5.2. The designed controllers will still work in stronger winds, although the response would be slightly different. This is shown in Paragraph 5.2.3.

Length of tether	**Offset (K)**	**Natural frequency (ω_n)**	**Damping coefficient (ζ)**
$5m$	$0.36m$	$0.48rad/s$	0.1
$10m$	$0.72m$	$0.35rad/s$	0.1
$15m$	$1.08m$	$0.28rad/s$	0.1
$20m$	$1.44m$	$0.25rad/s$	0.1

Table 5.3: Results of system identification tests for position controller (wind velocity = 3 m/s)

5.2.2 Root locus Design of Position Controllers

This section shows how different controllers were designed for four different linear models, according to four different tether lengths. These four controllers were designed according to the same control technique.

The first step was to transform the different second order linear transfer functions to their discrete equivalents. This was done by substituting the parameters of Table 5.2 into Equation 5.14 and use MATLAB to transform these equations to their discrete equivalents. The sampling period of the controller was chosen as $T_s = 1.0s$. This led to the four discrete linear transfer functions described in Equation 5.15 to Equation 5.18. The length of the tether is given in brackets next to the discrete linear transfer function.

$$G(z) = \frac{0.0044z + 0.0044}{z^2 - 1.75z + 0.972} \qquad (tether = 5m) \tag{5.15}$$

$$G(z) = \frac{0.0048z + 0.0048}{z^2 - 1.86z + 0.979} \qquad (tether = 10m) \tag{5.16}$$

$$G(z) = \frac{0.0046z + 0.0046}{z^2 - 1.91z + 0.983} \qquad (tether = 15m) \tag{5.17}$$

$$G(z) = \frac{0.0053z + 0.0053}{z^2 - 1.91z + 0.979} \qquad (tether = 20m) \tag{5.18}$$

The root locus plots of these four discrete linear transfer functions were used to design the controllers. The time domain specifications of the closed loop systems were chosen so that it would have a 1% settling time of $t_s = 10s$ and a damping coefficient (ζ) of 0.7.

The approximations of the time domain specifications for second order systems, as given in [27], were used to calculate the closed loop poles of these control systems. Equation 5.19 gives the location of the closed loop poles.

$$z_{CL} = 0.578 \pm j0.280 \tag{5.19}$$

The controllers can now be designed so that the specified closed loop characteristics could be obtained. All four controllers were designed according to the same technique: First of all an integrator was added to the system. Then the two open-loop poles were cancelled by adding a complex zero pair to the system. Another pole was added to the system so that the root locus moved through the position of the desired closed-loop poles. Finally the gain was adjusted so that the closed-loop poles were at the desired locations.

Figure 5.22 shows the root locus plot for the controller where the tether is $5m$ long.

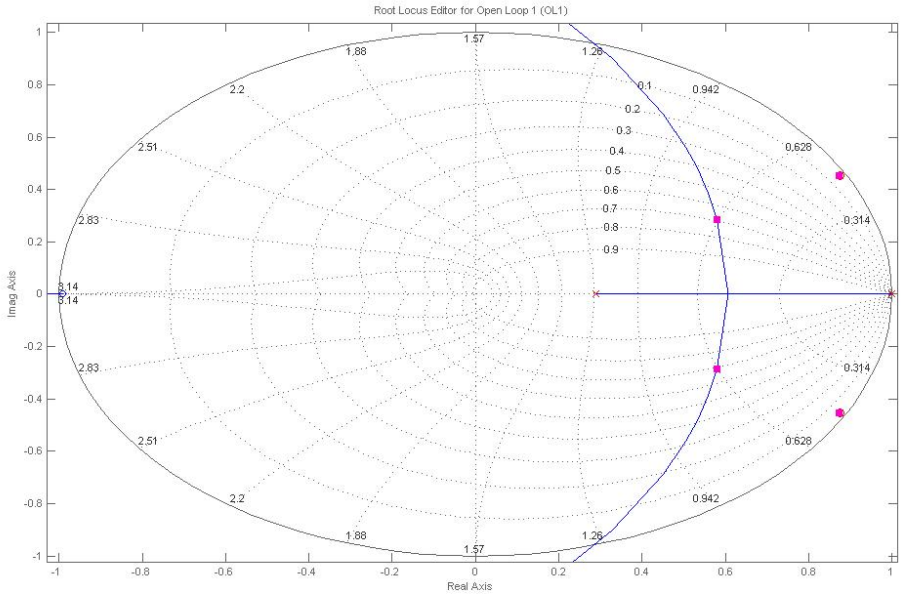


Figure 5.22: Root locus plot of position controller (tether = 5 m)

Equations 5.20, 5.21, 5.22 and 5.23 give the discrete transfer functions of the designed controllers. The length of the tether is given in brackets next to the equation.

$$D(z) = 1.1623\frac{z^2 - 1.75z + 0.972}{z^2 - 1.29z - 0.29} \qquad (tether = 5m) \tag{5.20}$$

$$D(z) = 2.10\frac{z^2 - 1.86z + 0.979}{z^2 - 1.287z + 0.287} \qquad (tether = 10m) \tag{5.21}$$

$$D(z) = 3.412\frac{z^2 - 1.91z + 0.983}{z^2 - 1.285z + 0.285} \qquad (tether = 15m) \tag{5.22}$$

$$D(z) = 3.712\frac{z^2 - 1.91z + 0.979}{z^2 - 1.30z + 0.30} \qquad (tether = 20m) \tag{5.23}$$

5.2.3 Simulation Results of Position Controllers

The designed controllers in Paragraph 5.2.2 had to be tested and verified in simulation before testing it in practice.

Figure 5.23 shows how the controller described in Equation 5.23 was implemented into the simulation of the airship. A zero order hold circuit was implemented to discretize the position output of the airship model. The position error was calculated and the output of the controller was limited and connected to the actuator inputs of the airship model. The length of the tether that corresponded to this position controller was $20m$.

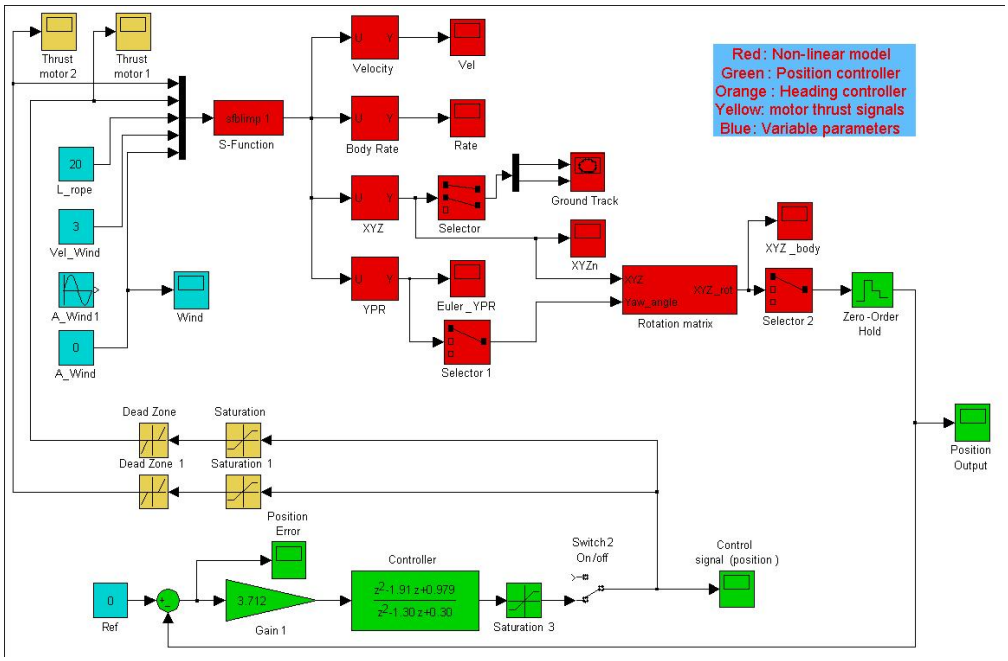


Figure 5.23: Simulation block diagram of position controller

Figure 5.24 shows how the position controller corrected the position error and tried to keep the airship at its original position where the tether was vertical. A constant wind of $1m/s$ was used in this simulation.

Figure 5.25 shows the control signal that was commanded to the actuators for the simulation in Figure 5.24. The unit of the control signal is given in Newton.

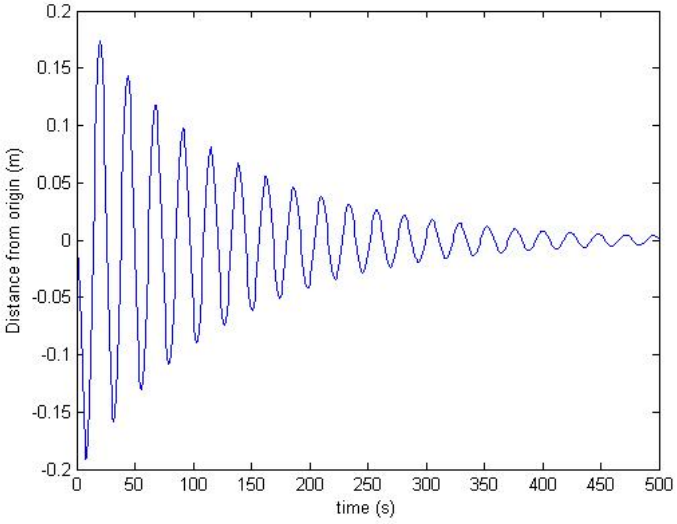


Figure 5.24: Simulation result of position controller in $1m/s$ wind ($tether = 20m$)

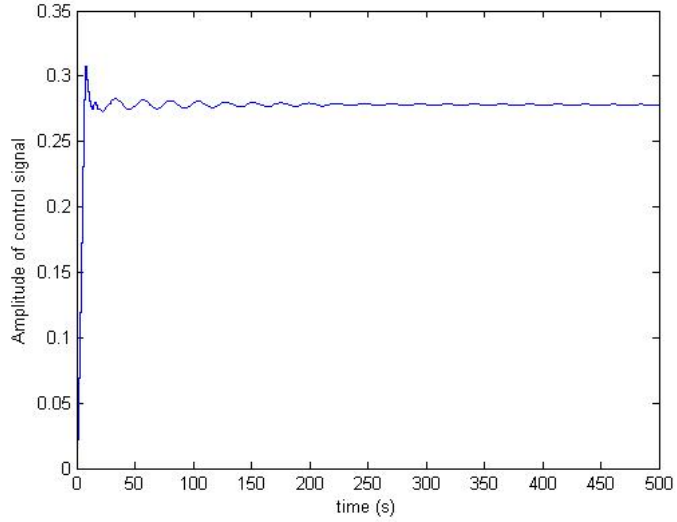


Figure 5.25: Control signal of simulation ($v_{wind} = 1m/s$)

Figure 5.26 shows how the same controller reacted in a stronger wind. The wind velocity was changed to $3m/s$ for this simulation.

Figure 5.27 shows the control signal that was commanded to the actuators for the simulation in Figure 5.26.

This confirms the statement that was made in Paragraph 5.2.2, that the designed controllers will still work in stronger winds, although the response would differ from the expected response which controller was designed for.

It is notable from Figure 5.24 and Figure 5.26 that it took a long time for the oscillations to be damped out completely. From Figure 5.25 and Figure 5.27 it is seen that the control signal quickly reached the desired amplitude that was needed to overcome the drag force of the wind. The controllers could be designed differently so that the oscillations will be damped out faster, but this

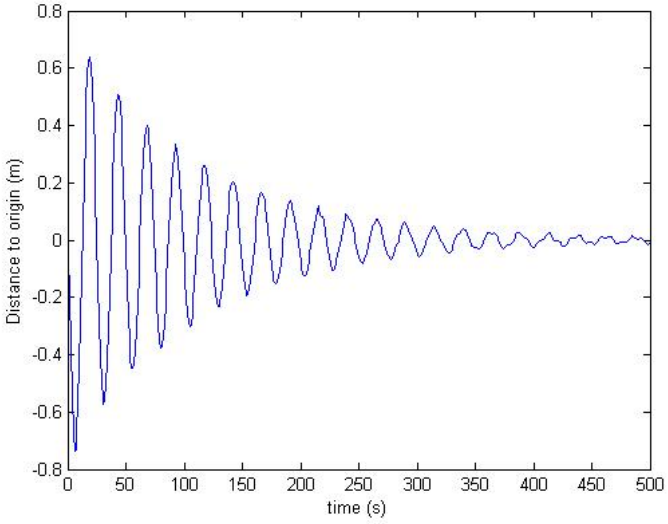


Figure 5.26: Simulation result of position controller in $3m/s$ wind ($tether = 20m$)

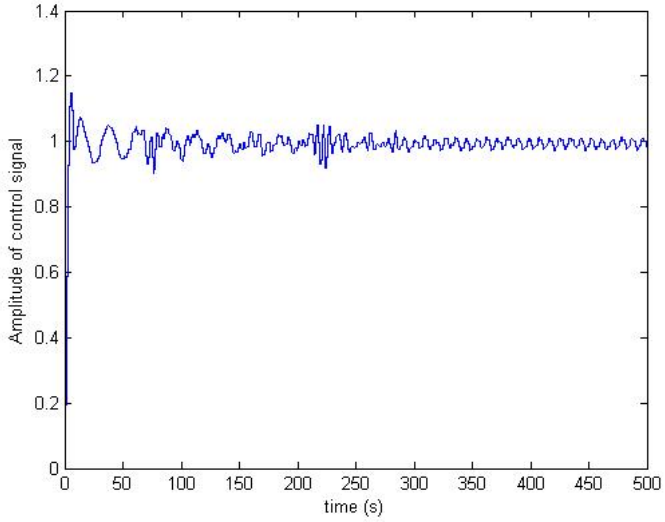


Figure 5.27: Control signal of simulation ($v_{wind} = 3m/s$)

will use much more control effort. The actuators that were used in this thesis are not extremely powerful. It is therefore better to keep the position controllers as it is for these specific actuators so that the control effort is kept within the capabilities of the actuators.

The other position controllers were all tested in simulation. The simulation results for all the controllers were similar to the results shown in Figure 5.24 and Figure 5.26.

5.2.4 Implementation of Position Controllers

In order to implement the designed controllers on a computer or a micro-processor, it is necessary to rewrite the discrete transfer functions of these controllers as difference equations. The method for writing a discrete transfer function as a difference equation is given in Paragraph 5.1.4.

The four difference equations that correspond to the four controllers are given in Equation 5.24 to Equation 5.27.

$$u_2(k) = 1.29u(k-1) - 0.29u(k-2) + 1.1623\big(e(k) - 1.75e(k-1) + 0.972e(k-2)\big) \qquad (tether = 5m) \tag{5.24}$$

$$u_2(k) = 1.287u(k-1) - 0.287u(k-2) + 2.10\big(e(k) - 1.86e(k-1) + 0.979e(k-2)\big) \qquad (tether = 10m) \tag{5.25}$$

$$u_2(k) = 1.285u(k-1) - 0.285u(k-2) + 3.412\big(e(k) - 1.91e(k-1) + 0.983e(k-2)\big) \qquad (tether = 15m) \tag{5.26}$$

$$u_2(k) = 1.30u(k-1) - 0.30u(k-2) + 3.712\big(e(k) - 1.91e(k-1) + 0.979e(k-2)\big) \qquad (tether = 20m) \tag{5.27}$$

These difference equations were all implemented in C++ in the Ground Station software. The software decides which controller to use, depending on the given length of the tether. The four controllers were designed for four specific tether lengths, but the Ground Station software will determine the controller which is closest to the specified length of the tether. For instance: If the tether length is specified as $9m$, the Ground Station software will use the controller that was designed for a tether of $10m$.

The control signal ($u_2(k)$) should not exceed the maximum control effort which the actuators can deliver to the system. Each actuator can deliver a forward thrust of approximately $2.1N$ to the airship as measured and shown in Figure 4.9. Therefore the control signals should be limited to the maximum and minimum values which can be applied to the actuators. If the control signal is limited to these values, it will also prevent integrator windup from occurring.

5.2.5 Practical Results of Position Controller

Only one of the position controllers could be tested practically. The reason for this is that the position controller needed to be tested in a wind free environment. The only indoor facility available limited the length of the tether to $5m$. The controller that corresponds to a $5m$ tether was therefore tested in this facility.

The test was done in the following manner: The airship started from its original position with an arbitrary heading. The heading controller was switched on to maintain the current heading. The position controller was then commanded to step the airship $1m$ forward from its initial position.

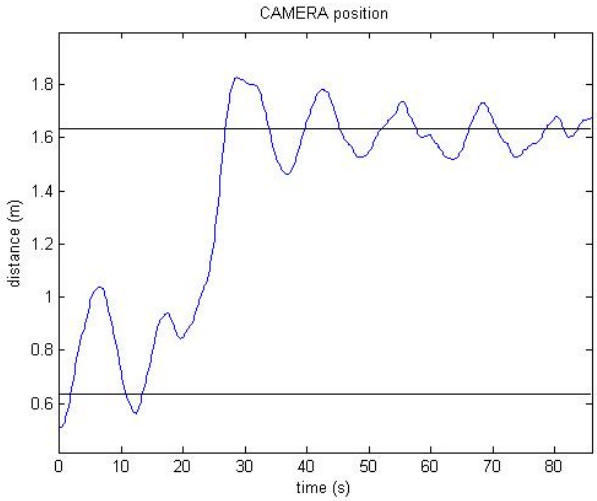


Figure 5.28: Step response of position controller

Although there was no wind present when this test was done, the position controller had to overcome the pulling force in the tether in order to settle the airship at its new position.

Figure 5.28 shows how the airship was practically step by $1m$ towards a new position and how the oscillations were damped out.

Figure 5.29 shows how the heading of the airship was kept constant during this practical test.

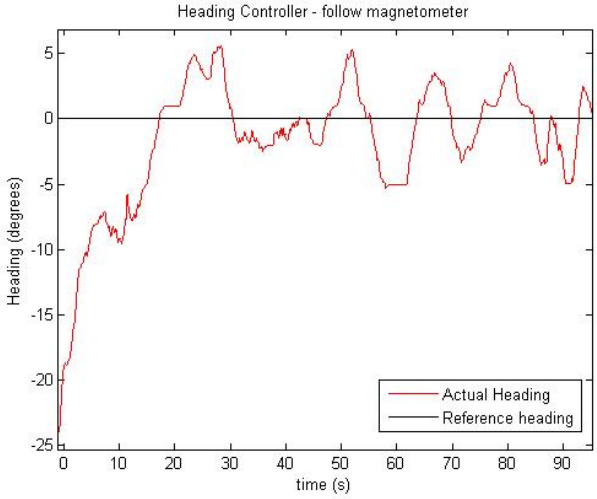


Figure 5.29: Maintaining a specified heading in order to test position controller

Figure 5.30 shows the control signal that was applied to the actuators during this practical test.

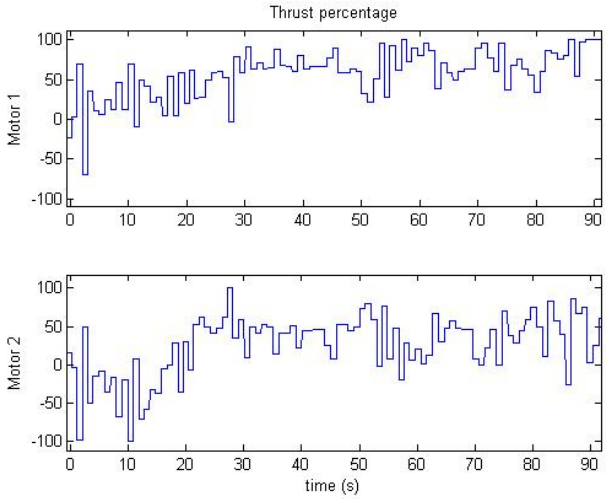


Figure 5.30: Control signal of practical test using both controllers

5.3 Combining the heading controller and position controller

The heading controller and position controller must work together in order to keep the airship stationary.

The control signal of the two controllers should be combined in such a way that the heading and position of the airship is controlled simultaneously. The combined control signals that are commanded to each actuator are given in Equation 5.28 and Equation 5.29 respectively. The sum of the two control signals is limited according to the maximum and minimum control effort that the actuator possesses.

$$u_{motor_1}(k) = u_1(k) + u_2(k) \tag{5.28}$$

$$u_{motor_2}(k) = -u_1(k) + u_2(k) \tag{5.29}$$

$u_1(k)$ refers to the control signal of the heading controller. $u_2(k)$ refers to the control signal of the position controller. Note that the sign of the control signal of the heading controller differs for the two motors. This was done to generate a yaw-moment.

Let's look at a practical situation where the airship needs to be controlled in a wind of $1m/s$: Figure 5.25 shows that the position controller will command a forward thrust of approximately $0.3N$ in order to oppose the drag force of a $1m/s$ wind. $0.3N$ is 14% of the maximum forward thrust that can be generated by a motor. Let's say that the heading controller was applying a $0.6Nm$ yaw-moment to the airship at the same time. $0.6Nm$ refers to 33% of the maximum yaw-rate. The motor that applies the negative thrust should be applying 33% of its maximum thrust to the airship. The other motor should only apply 23.1% of its maximum thrust in the forward direction, according to the difference in efficiency of the propellers.

Equations 5.28 and 5.29 were used to obtain the thrusts that should be commanded to each motor. The result is shown in Equations 5.30 and 5.31 respectively.

$$u_{motor_1}(k) = 23.1\% + 14\% = -19\% \tag{5.30}$$

$$u_{motor_2}(k) = -33\% + 14\% = 37.1\% \tag{5.31}$$

These respective thrusts should be commanded to the motors. The result will have the desired effect of turning the airship in the correct direction, while continuing to oppose the drag force generated by the wind.

Figure 5.31 shows the blockdiagram of the complete airship system with both the heading controller and position controller functioning simultaneously.

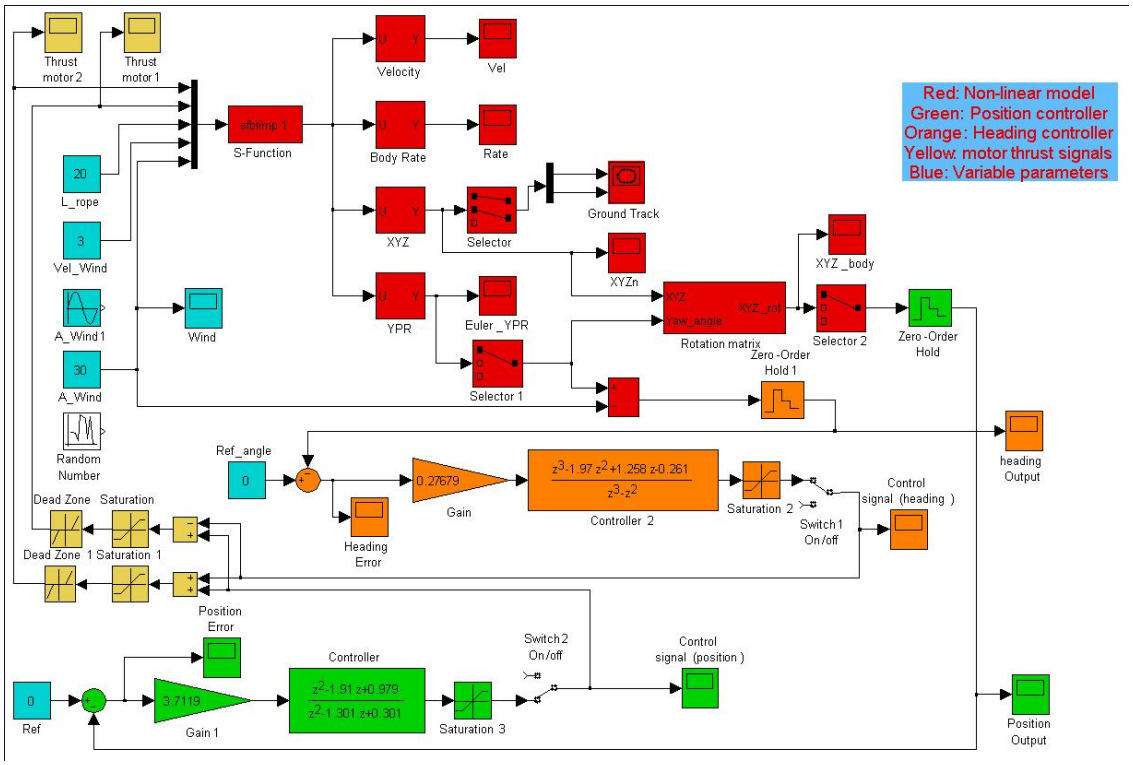


Figure 5.31: Blockdiagram of complete simulation

A simulation to illustrate the simultaneous working of the controllers was performed. A constant wind of $3m/s$ at an initial heading of 30° was applied in the simulation. The length of the tether was $20m$ for this simulation. The $T_s = 1.0s$ heading controller was used in this simulation. The position controller for a $20m$ tether was used. The results of the simulation are shown in Figures 5.32 to 5.36.

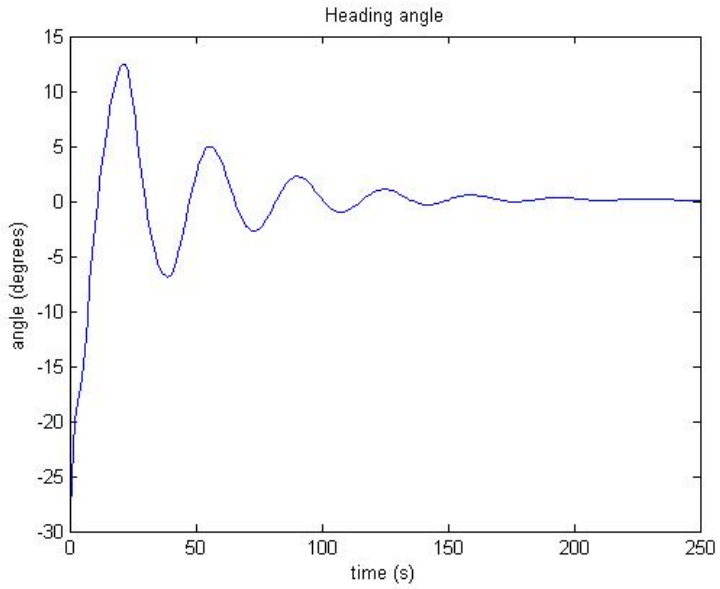


Figure 5.32: Heading angle output (constant wind direction)

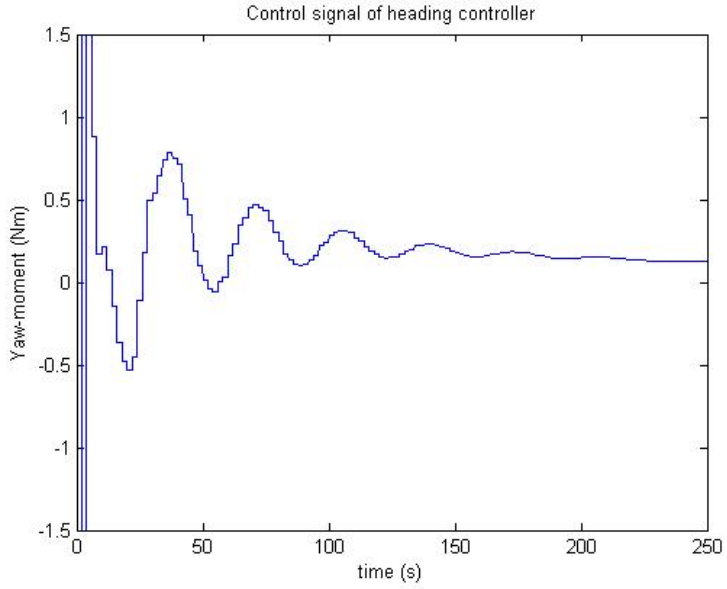


Figure 5.33: Control signal of heading controller (constant wind direction)

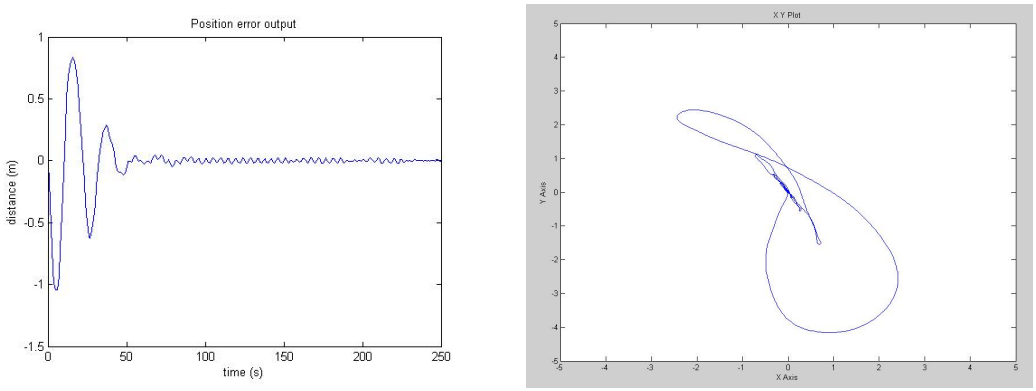


Figure 5.34: Position output and Ground Track of airship

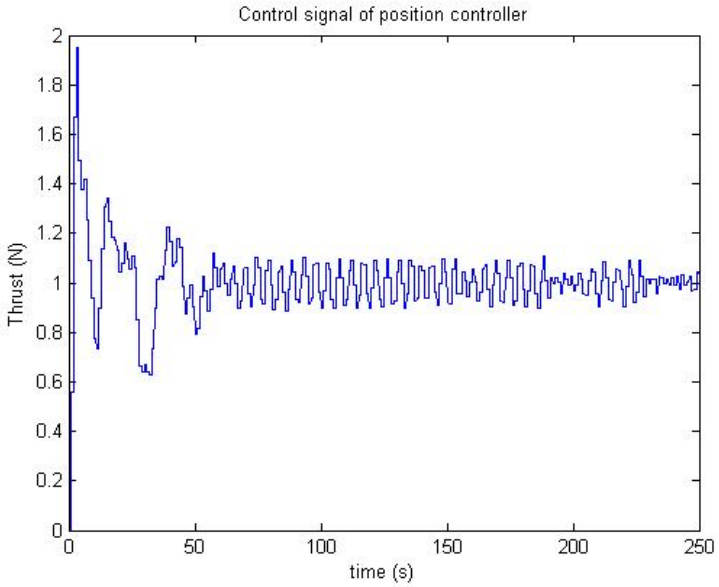


Figure 5.35: Control signal of position controller (constant wind velocity)

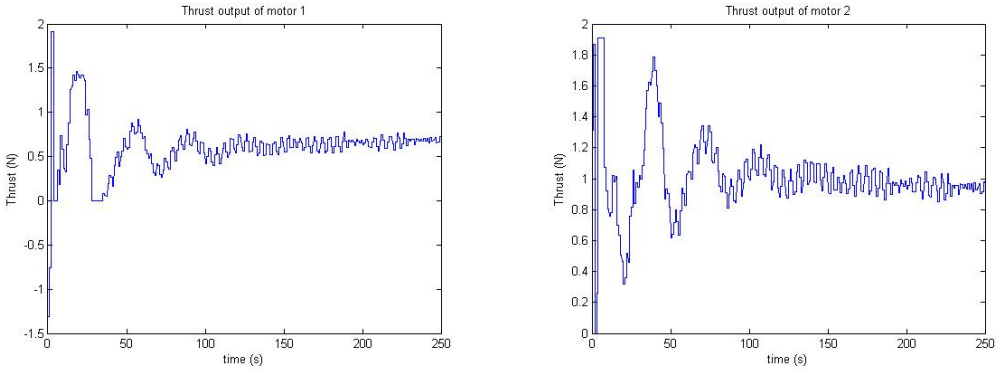


Figure 5.36: combined control signals (constant wind)

The simulation can be improved further by adding variances to the wind velocity and wind direction. This will create a more realistic wind model which will illustrate the robustness of the control system.

For the next simulation a random number was added to the constant wind velocity of $3m/s$. The random number has a mean value of zero and a magnitude of $1m/s$. The sampling rate of the random number is $2s$. The wind direction is also varied by a random number. This random number also has a mean value of zero but has a magnitude of 10°. The sampling rate of the random number is $30s$ in this case.

Figures 5.37 to 5.40 shows the results of this simulation.

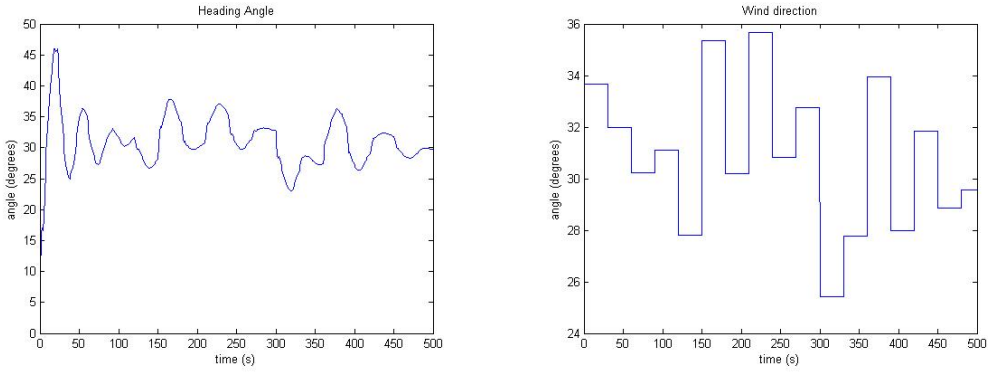


Figure 5.37: Heading angle output with variable wind direction

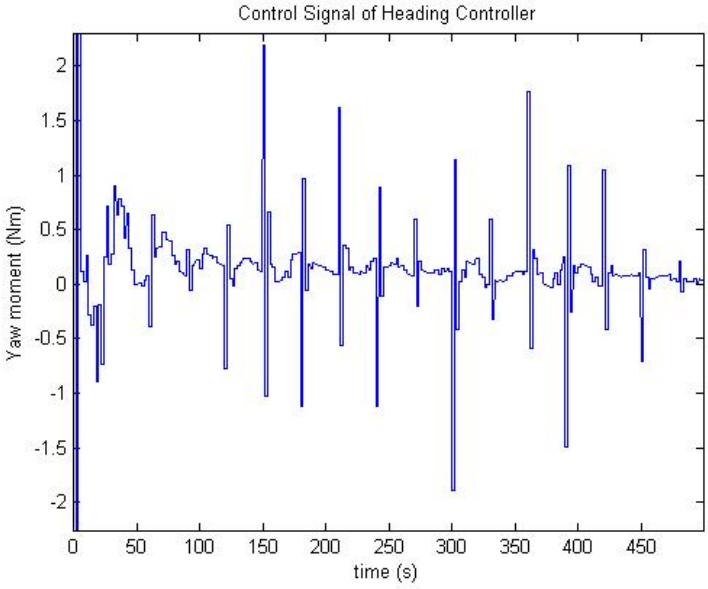


Figure 5.38: Control signal of heading controller (varied wind)

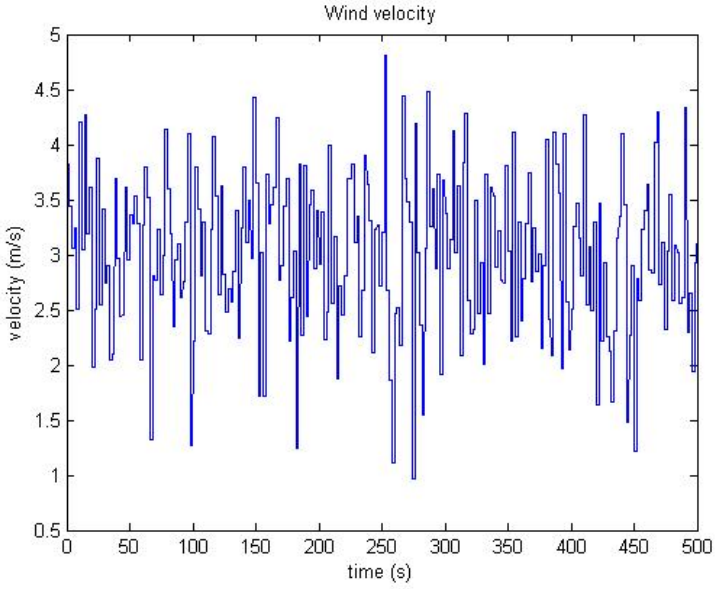


Figure 5.39: Wind velocity

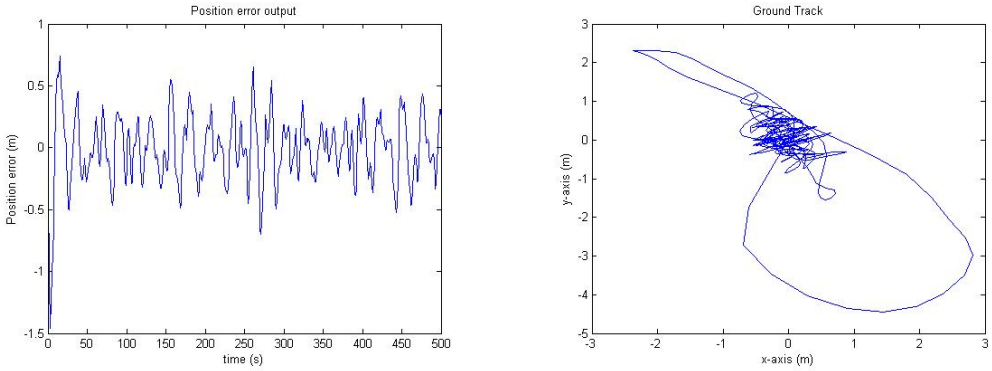


Figure 5.40: Position output and Ground Track of airship (varied wind velocity)

The simulated results show that the control system works in low winds. The simulation suggests that the control system works best when a constant wind is applied, but the accuracy doesn't decline too much when the wind is varied slightly.

The final test would demonstrate the working of the control system outdoors, in practical wind conditions. This proved to be a problem. Although the simulation suggests that the airship should be controllable in a wind of up to $4m/s$, this was not the case when testing the system outdoors.

The airship was tested practically in a wind that measured between $0m/s$ and $4m/s$ but the controllers were unable to dampen the oscillations of the airship's movements. This is due to the actuators not being powerful enough to compensate for the ferocious movements of the airship in an inconsistent and gusty wind. The actuators are only capable of rotating the airship at a maximum yaw rate of $3°/s$ which proves to be insufficient when the airship operates in a practical wind. Figure 5.41 shows the effect of a practical wind on the heading of the airship. This figure shows how the wind makes the airship rotate at up to $45°/s$.

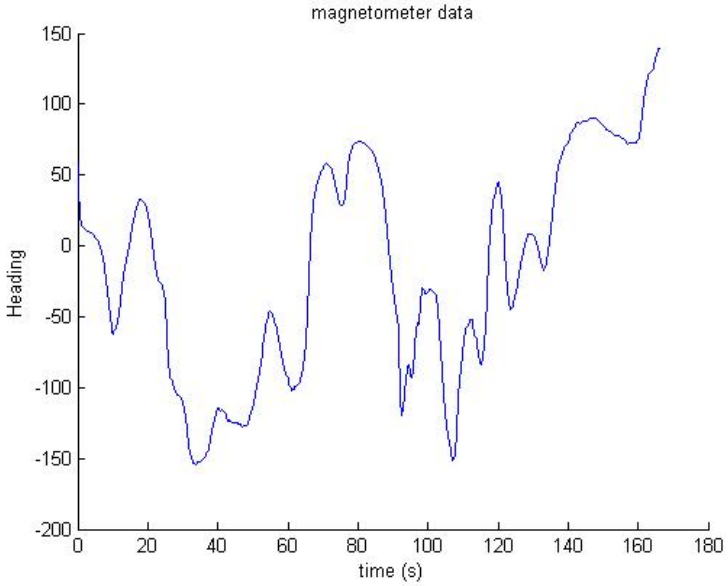


Figure 5.41: Effect of practical wind on airship's heading

Unfortunately it was impractical to redesign the airship system with stronger actuators at this point. The results obtained from testing the control system indoors illustrates that the problem is not with the control system itself, but with the actuators. The system should work very well with stronger actuators that are capable of rotating the airship at rates of up to $45°/s$.

Another flight test was attempted in a wind of less than $1m/s$. The heading controller manages to control the airship's heading, to follow the measured wind direction, with reasonable accuracy during this test. This is shown in Figure 5.42.

Figure 5.43 shows the measured wind during this test. The accuracy of the heading controller decreased even though the measured wind speed is very low.

This test confirms that the controllers are working, but that the actuators are limiting the performance of the control system in practical wind conditions.

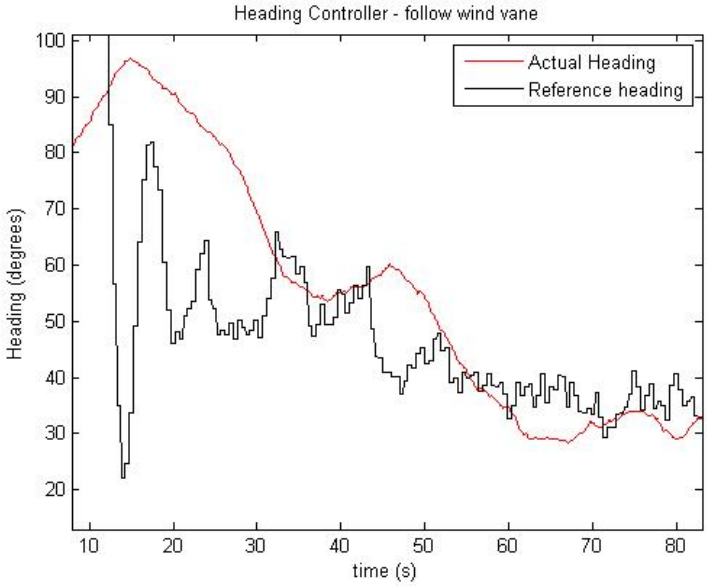


Figure 5.42: Practical results of heading controller in light wind

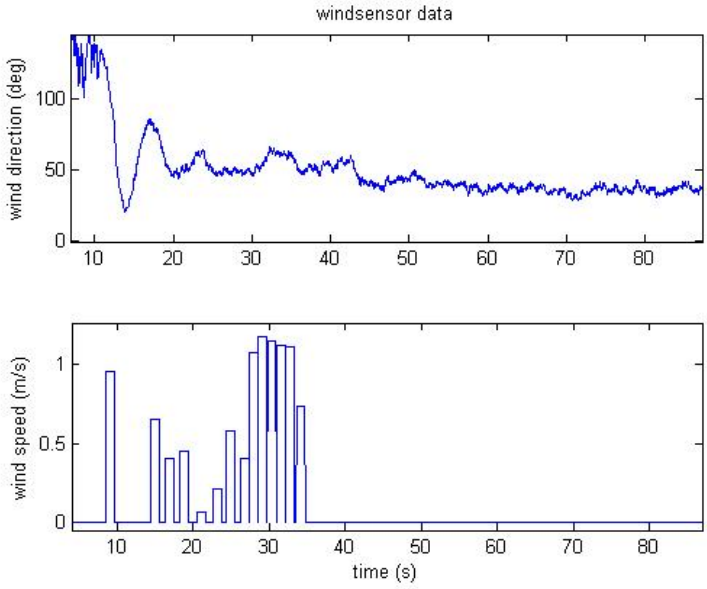


Figure 5.43: Measured wind for practical flight test

Chapter 6

Conclusion

This chapter reflects back on the required objectives identified at the beginning of the project. The extent to which these objectives were reached is discussed.

The first objective was to design and build a hardware platform with actuators that could be used to control the heading and position of an airship. The size and weight of the hardware are some of the many constraints which needed to be dealt with. The hardware specifications required the use of certain sensors and electronic equipment, like wind sensors and a GPS receiver unit.

A light-weight hardware platform was designed and built with all the necessary sensors on board the platform. A microprocessor was used to sample the sensor measurements. The microprocessor was also used to control the actuators. Two brushless DC motors were used as actuators. The hardware platform, including the actuators, is powered from the ground through a tether. This was done to minimise the weight of the hardware platform.

Software was developed for the microprocessor to ensure that the sensor measurements are sampled correctly and frequently enough. All the sensors were calibrated and tested separately. The sensors use a variety of communication protocols like I^2C and serial communication which needed to be dealt with. Some sensors are analog devices while others are digital. All these different devices were configured to work together in the hardware configuration.

The second objective was to develop a simulation of the static and dynamic model of the airship. The static and dynamic model was developed mainly by the use of the PhD thesis of Yuwen Li [3]. The static and dynamic model was implemented in a Simulink simulation. Wind was modelled as an input to the system. The physical airship system was compared to the simulation results and it was found that the developed airship model closely resembles the characteristics of the physical airship system.

The third objective was to incorporate the sensors and the actuators on the gondola into a system that could control the heading and position of the airship. A ground station was developed for the purpose of monitoring the movements of the airship. The hardware platform and ground station system were integrated so that the airship could be monitored and controlled from the ground station. The ground station was used to collect and store data during system identification tests.

The data that was collected during system identification tests were used to create linear heading and position models of the airship. The linear models were used to design a heading controller as well as a position controller for the airship. The controllers were implemented in simulation, as well

as practically in the ground station computer. The simulation results verify that the designed linear controllers are capable of controlling the non-linear airship model.

The controllers were demonstrated practically in conditions where no wind was present. This demonstration showed that the controllers respond appropriately for various step commands.

It was attempted to demonstrate the practical working of the airship system in the presence of wind, but it was found that the wind was too gusty and the actuators were not powerful enough to efficiently counter the disturbances of the wind. The use of stronger actuators should solve this problem, but stronger actuators will require more power. It was impractical to redesign the hardware platform to incorporate stronger actuators at this point. Therefore the control system was only demonstrated in low winds.

This project shows that it is possible to control an airship for it to remain stationary. This could lead to future projects where airships could be used as geostationary platforms and Stratolites. The literature review of Paragraph 1.2 looked at this possibility. There were however some limitations to this project which influenced the extent of the research done on the development of Stratolites. These limitations are discussed in Paragraph 6.1.

Paragraph 6.2 gives a summary of the achievements of this project.

Paragraph 6.3 gives some useful recommendations that should be considered for future projects.

6.1 Limitations

The following list contains the limitations that existed in this project:

- The airship used during this project is reasonably small. A Stratolite would need a much bigger airship to carry all the equipment on board the airship. Batteries and solar panels should be fitted to a Stratolite, which will make it even heavier.

- The plastic material from which the hull was made is not dense enough. The small hydrogen or helium particles escape through the hull. It would be difficult, if not impossible, to prevent the gas particles from escaping completely, but the best possible material should be used. This would allow the airship to have longer flight endurance.

- The actuators used is only effective in low winds. The use of stronger actuators with higher thrust capability would be able to control the airship in more realistic wind conditions. Stronger actuators will require even more power, which will further increase the number of batteries and solar panels needed.

- The limitations of the payload resulted in the airship to be powered from the ground through a tether. This could be problematic if the length of the tether is increased. A Stratolite should be powered by solar power and batteries, without the use of a tether. The extra weight which this airship should be able to carry would have a dramatic effect on the size of the airship.

- The airship has no actuators or control surfaces to control the altitude of the airship. A Stratolite should be able to descend back to earth by either using an actuator or by releasing the lifting gas through a controlled valve. This should be considered when developing a Stratolite.

- The airship used in this project would not be able to withstand any bad weather conditions. Although a Stratolite will operate above the jet stream in the upper atmosphere, it would still need to be much more robust than the airship used in this project. The gondola of a Stratolite would need to be waterproof and all the electronics should be carefully protected from any harsh weather conditions.

- The Camera Tracker used as a position sensor in this project would only be effective over a couple of meters and only when the LED on the airship is brighter than the surrounding light. A Stratolite would be too far away from the ground to use such a position sensor. The GPS receiver which is available in this project would then be used as the position sensor on the Stratolite.

- The wind direction sensor used in this project is not very effective when it is situated on the gondola. The wind vane was moved to the ground and was connected to the airship with a long cable. The accuracy of the wind vane measurements decreased because of the long cable. The wind direction vane used is very sensitive to wind gusts. Some of these wind gusts make the wind vane turn through 360° in a short period. This is unacceptable and makes it increasingly difficult to use this sensor to control the airship's practical movement in the wind. A better sensor should be developed for the specific application on a Stratolite. This sensor should be fitted onto the Stratolite and should be able to accurately measure the wind direction.

6.2 Achievements

The following goals were achieved in this project:

- An accurate model of the airship system with implemented controllers was developed in Simulink. The controllers were tested and proven to work in simulation.

- A practical airship system was designed and built according to the specifications.

- System identification tests were performed on the airship system to acquire linear models for the heading and position of the airship. These linear models were used to design a control system for the airship system. The control system consists of a heading controller and a position controller.

- The control system was implemented and tested practically. The control system performs well in practise as long as there is no wind. It was difficult to test the control system in practical wind conditions. This was due to the inefficiency of the wind direction sensor and due to an under actuated system. This problem can be resolved by using a more reliable wind direction sensor and stronger actuators.

6.3 Recommendations

The following list contains valuable information that could be extremely insightful for future projects:

- The design of the hull of the airship is extremely important. This could be a complete study field on its own. The shape, size and material of the hull should be specifically designed for the requirements of the airship. This would be extremely important when developing a Stratolite.
- The actuators should be larger than the ones that were used in this project. This would give the airship more available control effort which will allow the airship to operate in stronger winds.
- A study should be made on the size of the actuators that would be needed to efficiently control different sizes of airships in practical wind conditions. The amount of power needed to power stronger actuators should be examined. This should be done to determine the amount of batteries that should be used and what the maximum endurance of such an airship would be.
- Extra actuators at the fins could be considered to increase the available yaw-rate.
- The actuators can alternatively be moved further away from each other to increase the available yaw-rate.
- The hardware of the airship should be fitted with a pitot tube so that the altitude and airspeed could be measured more accurately.
- The control system of a Stratolite should be implemented on the microprocessor on the Stratolite rather than controlling it from a ground station. The ground station should mainly be used to monitor the Stratolite.

References

[1] Avenant, G.: *Autonomous Flight Control System for an Airship.* Master's thesis, University of Stellenbosch, 2009.

[2] Jacobs, M.: Stratospheric platforms for telecommunication applications. Presentation, March 2004.

[3] Li, Y.: *Dynamics Modeling and Simulation of Flexible Airships.* Ph.D. thesis, McGill University, Montreal, 2007.

[4] Li, Y. and Nahon, M.: Simulation of airship dynamics. *AIAA Modeling and Simulation Technologies Conference and Exhibit*, vol. AIAA 2006-6618, 2006.

[5] *Datasheet, 2-Axis Compass with Algorithms (HMC6352).* Honeywell, 2006.

[6] *ANTARIS 4 System Integration Manual, Doc No GPS.G4-MS4-05007*, 2007.

[7] Schmidt, D., Stevens, J. and Roney, J.: Near-space station-keeping performance of a large high-altitude notional airship. *AIAAA 2006-6510*, 2006.

[8] Khoury, G.: *Airship Technology.* Cambridge University Press, 1999.

[9] 21st century airships, "canadian airship demonstration wows american military". Media Release, June 2004.

[10] Mueller, J. and Paluszek, M.: Development of an aerodynamic model and control law design for a high altitude airship. *American Institute of Aeronautics and Astronautics*, 2004.

[11] Hima, S. and Bestaoui, Y.: Motion generation on trim trajectories for an autonomous underactuated airship. *Proceedings of the 4th international airship convention & exhibition*, 2002.

[12] Elfes, A., Bueno, S. and Bergerman, M.: Development of an autonomous unmanned remote monitoring robotic airship. *Journal of the Brazilian Computer Society, ISSN 0104-6500*, 2006.

[13] Ashford, R. and Bely, P.: High altitude aerostats as astronomical platforms. Tech. Rep., TCOM, 1995.

[14] Elfes, A. and Montgomery, J.: Autonomous flight control for a titan exploration aerobot. Tech. Rep., Jet Propulsion Laboratory, Pasadena, CA 91109, 2005.

[15] Bijker, J.: *Development of an Attitude Heading Reference System for an Airship.* Master's thesis, University of Stellenbosch, 2006.

[16] Aerovironment inc. website, http://www.avinc.com/uas. June 2009.
Available at: http://www.avinc.com/uas

[17] Hindle, S. and Wierzbanowski, T.: Skytower high altitude platform stations, enabled by av global observer unmanned aircraft system. In: *ITU CPM Update*. 2007.

[18] Nebylov, A.: *Automatic Control in Aerospace 2004*. Elsevier IFAC publications, 2004.

[19] Yokomaku, Y.: Overview of stratospheric platform airship r&d program in japan. In: *Stratospheric Platform Systems Workshop*. 2000.

[20] Kantora, G. and Wettergreena, D.: Collection of environmental data from an airship platform.

[21] Etkin, B.: *Dynamics of Flight: Stability & Control*. Wiley, New York, 1996.

[22] Hopkins, E.: A semiempirical method for calculating the pitching moment of bodies of revolution at low mach numbers. Tech. Rep., Ames Aeronautical Laboratory, 1951.

[23] Gomes, S.: *An Investigation of the Flight Dynamics of Airships with Application to the YEZ-2A*. Ph.D. thesis, Cranfield Institute of Technology, 1990.

[24] *Datasheet, dsPIC30F4011/4012 : High-Performance, 16-bit Digital Signal Controllers*, 2008.

[25] *Datasheet, RCB-4H ANTARIS 4 Programmable GPS Receiver Board with SuperSense*, 2007.

[26] *16-Bit Language Tools Libraries*. Microchip, 2005.

[27] G.F. Franklin, J.D. Powell, M.W.: *Digital Control of Dynamic Systems*. Addison-Wesley, 1998.

Appendix A

Electronic Circuit Board

The electronic circuit board diagram is given in Figure A.1:

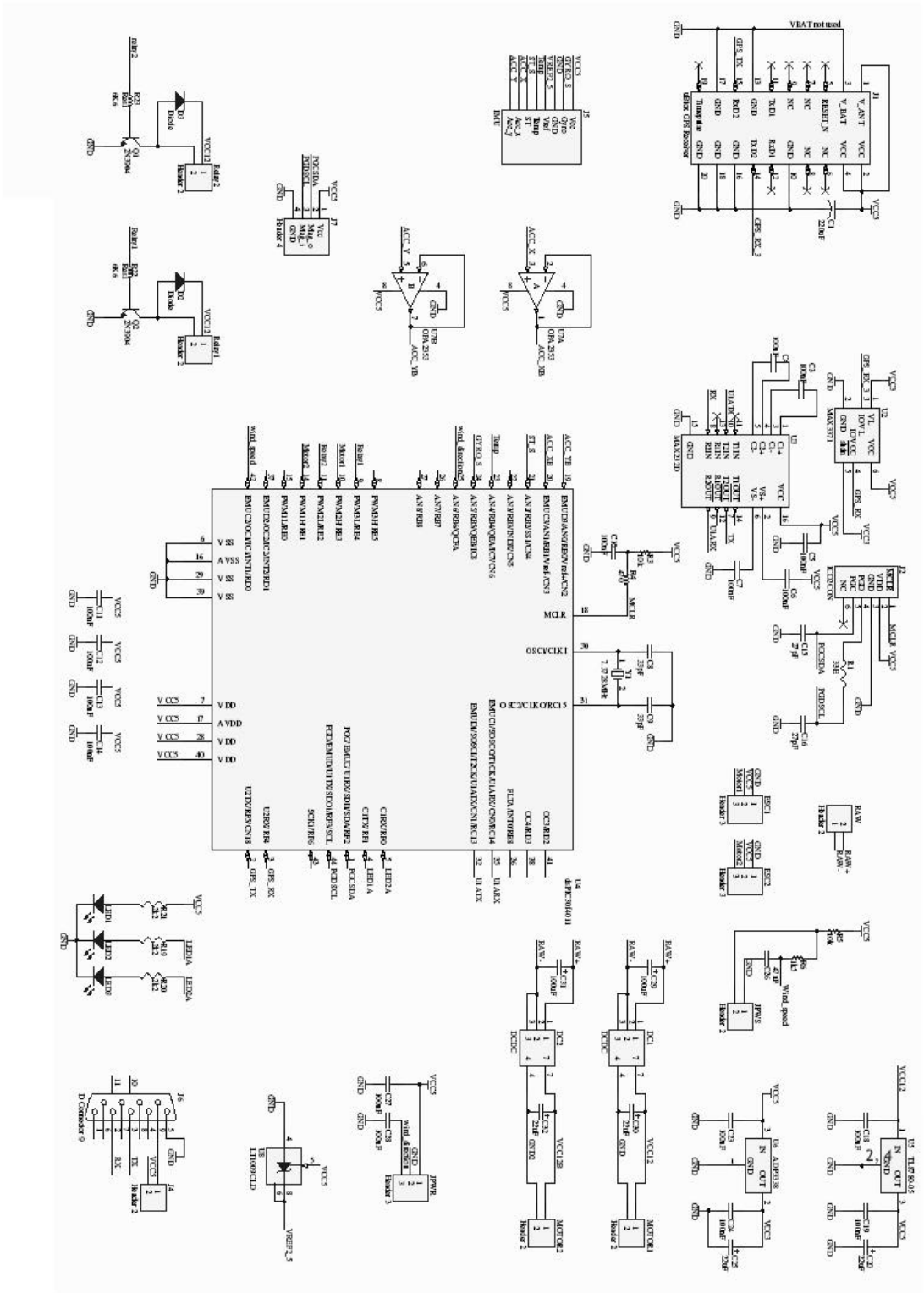


Figure A.1: Circuit diagram of HAP electronics board

Appendix B

System Block Diagram

This block diagram in Figure B.1 illustrates how the airship, ground station and camera tracker is connected. The various data packets that is transferred between the various systems is also shown.

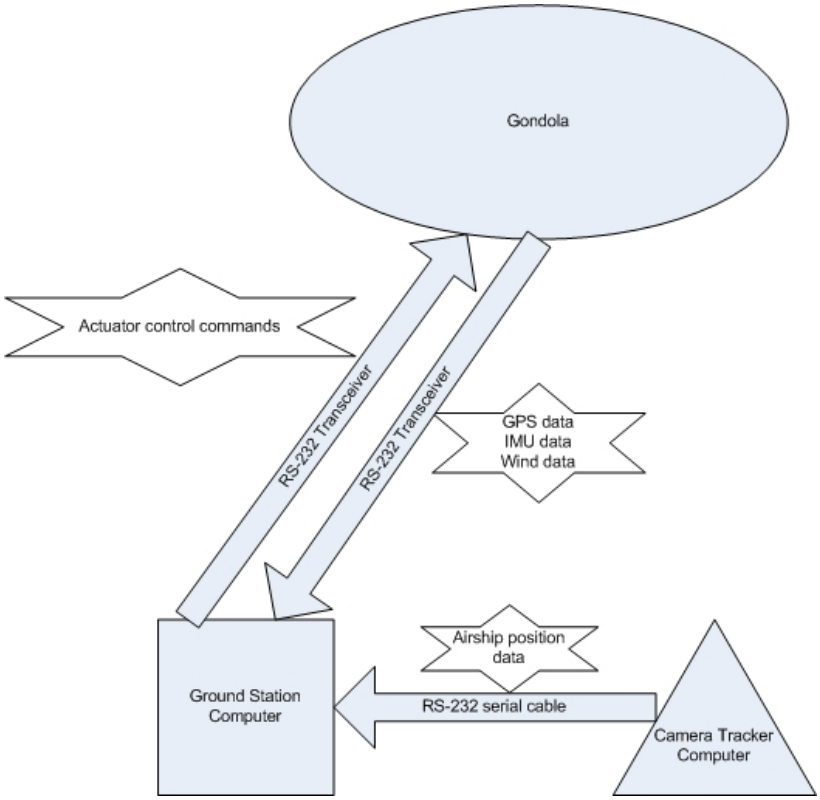


Figure B.1: Block diagram of complete system

Appendix C

Photos of setup during flight tests

The following photos are given as an illustration of how the airship system was set up and tested during various flight tests:

Figure C.1: Ground station setup

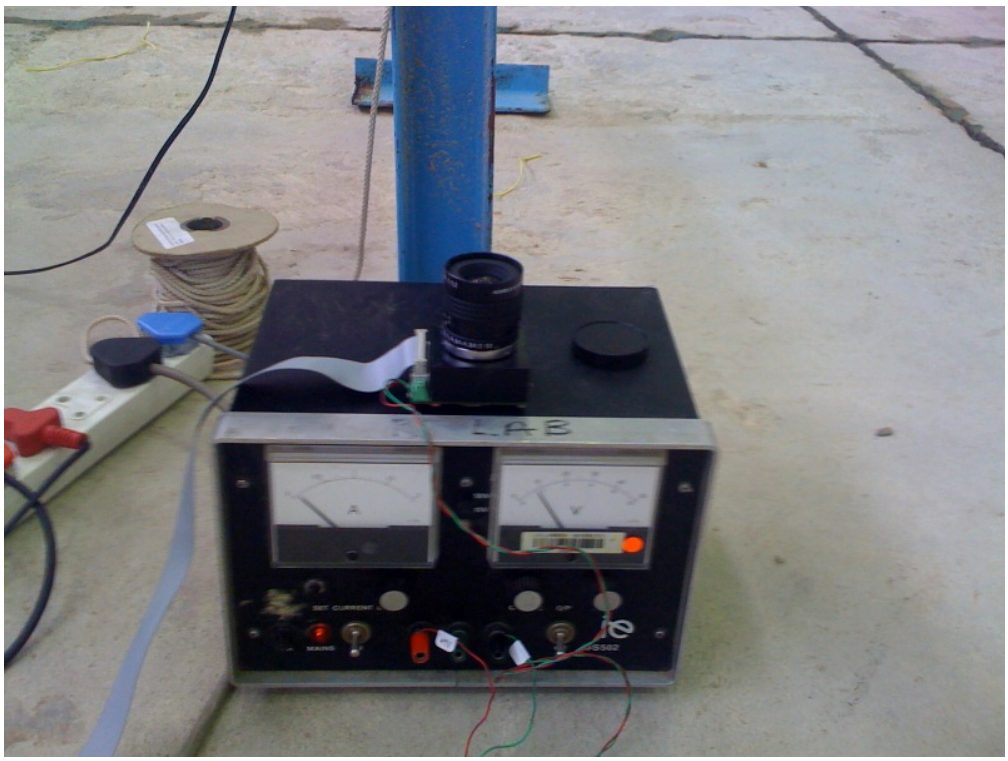


Figure C.2: Camera Tracker and power supply setup

Figure C.3: Airship setup with fan during system identification tests

Figure C.4: Tether connected to airship

Appendix D

Safety Precautions

The following safety precautions should be considered before any tests are done:

- Make sure there is no source of open flames that could ignite the hydrogen inside the hull of the airship. This is not an issue if helium was used to fill the hull.
- Be careful when electrical power supplies are switched on. Make sure that the voltage and current settings are correct before connecting the conductors to the power supplies. Make sure that the conductors are connected correctly before switching the power supplies on.
- Make sure that there is a safety officer present at all times.
- Beware of static electricity. Static electricity could destroy electronic components.
- Make sure that no object gets in the way of the propellers.
- Report any incidence of malfunction or any accidents to a supervisor.